Universidade de Aveiro Departamento de Engenharia Mecânica
2017

André Filipe Martins Antunes

Análise energética de sistemas de abastecimento de água: previsão dos consumos recorrendo a técnicas de inteligência artificial

Energetic Analysis of water supply systems: demand forecasting using artificial intelligence techniques

Universidade de Aveiro Departamento de Engenharia Mecânica
2017

André Filipe Martins Antunes

Análise energética de sistemas de abastecimento de água: previsão dos consumos recorrendo a técnicas de inteligência artificial

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor António Gil d'Orey de Andrade Campos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro

Apoio financeiro dos projetos UID/EMS/00481/2013-FCT e CENTRO-01-0145-FEDER-022083

Aos meus pais. À Filipa.

o júri

presidente

Prof. Doutora Mónica Sandra Abrantes de Oliveira Correia
Professora Auxiliar, Universidade de Aveiro

Prof. Doutora Leonor da Conceição Teixeira
Professor Auxiliar, Universidade de Aveiro

Mestre Miguel da Silva Oliveira
Assistente Convidado, Universidade de Aveiro (coorientador)

agradecimentos

A realização do presente trabalho não teria sido possível sem o apoio dos membros do grupo de investigação Water-GRIDS do DEM. A vocês o meu sincero obrigado, em especial ao meu orientador Gil Campos e ao coorientador Miguel Oliveira.

palavras-chave

Previsão de consumos de água, aprendizagem automática, sistemas de abastecimento de água, empresas fornecimento de água.

resumo

Hoje em dia, grande parte das empresas fornecedoras de água gere a sua operação com base na procura instantânea da rede, o que significa que a utilização dos equipamentos é condicionada pela procura imediata de água. Os reservatórios das redes são abastecidos recorrendo a bombas que são acionadas quando a água atinge o limite mínimo e desativadas quando esta atinge o limite máximo. Basear esta gestão na procura futura permite utilizar o equipamento de bombagem quando a energia elétrica é mais barata, ao tirar vantagem da tarifa elétrica em vigor, resultando numa diminuição de custos para a empresa.

A previsão de consumos a curto prazo é um passo fundamental no apoio à decisão referente à gestão da operação dos equipamentos. Para isso, uma série de metodologias de previsão são implementadas e analisadas em Python. Alguns métodos de *machine learning*, como redes neuronais, *random forests*, *support vector machines* e *k-nearest neighbours*, são avaliados usando dados reais de duas empresas fornecedoras de água portuguesas. Além disso, a influência de fatores como a meteorologia, sazonalidade, quantidade de dados usados no treino, e janela temporal das previsões também são testadas. Os resultados são validados e comparados com aqueles alcançados pelo ARIMA com recurso a benchmarks.

keywords

Water demand forecast, machine learning, water supply systems, water utilities.

abstract

In current days, a large number of water utilities manage their operation on the instant water demand of the network, meaning the use of the equipment is conditioned by the immediate water necessity. The water reservoirs of the networks are filled using pumps that start working when the water level reaches a specified minimum, stopping when it reaches a maximum level. Shifting the focus to management based on future demand allows to use the equipment when energy is cheaper, taking advantage of the electricity tariff in action, thus bringing significant financial savings over time.

Short-term water demand forecasting is a crucial step to support decision making regarding the equipment operation management. For this purpose, forecasting methodologies were implemented and analyses in Python. Several machine learning methods, such as neural networks, random forests, support vector machines and k-nearest neighbours, are evaluated using real data from two Portuguese water utilities. Moreover, the influence of factors such as weather, seasonality, amount of data used in training and forecast window are also tested. The results are validated and compared with those achieved by ARIMA using benchmarks.

CONTENTS

Energetic analysis of water supply systems: demand forecasting using artificial intelligence techniques

1. Context

1.1. Introduction

During pre-history, mankind only lived next to fresh water sources (rivers, lakes, springs). However, he realized he could capture rain water and store it. The settlement of large populations in water abundant areas was possible due to the construction of irrigation canals for agriculture and aqueducts, allowing water from multiple springs to be directed to aggregates of people and big cities. With the Industrial Revolution the water got a new application, both as a process ingredient and as a refrigerant for the machines. Nowadays, there is immediate access to water in almost any part of the world. However, there exists a global necessity to preserve the hydric resources and to employ the energy needed to water usage and distribution in a more efficient way. It is then necessary to find strategies that assure minimal water and energy waste, on high level networks (collection, treatment, storage) as well as on low level networks (from storage to delivery) [1].

1.2. The water sector

The contemporary water distribution networks, some of high complexity, make the connection between the source and the consumer, assuring water availability in quantity and quality. These include all the equipment and processes from collection to consumption, transport, filtration and storage [2]. The creation and maintenance of conditions that assure that the supply satisfies the demand at all times is the water utilities' job.

1.2.1. Water supply systems

Most water supply systems (WSS) are based on more than one source, which can be underground, spring, glacier, ocean, rain, or others. The water from the different sources can be joined together before distribution or each source can provide water for its own sub-network. It is usually found a combination of these solutions to solve the problems of seasonal scarcity and diminish the costs that increase with the distance and irregularity of the terrain between sources. After collection, the water is submitted to mechanical and chemical treatments that prepare it to consumption.

There is currently a concern with both the quality and quantity of the consumed and wasted water. Either for scarcity or economic reasons, measures must be taken to decrease the resources spent in water treatment. With this in mind it is ideal to collect the minimum amount of water that satisfies the customer's needs, assuring the longevity of the cleanest water sources.

The reservoir (tanks) existent in the networks allow the storage of water in the periods when it is less needed making it available for the periods when it is more needed and for eventual emergency occurrences. Additionally, a good location of the tank may mean enough pressure and no pumping is needed. This isn't always the case. For this reason it is common to find

pumps between tanks and consumers, and even between sources and tanks, allowing the water to be stored at a superior height. However, these reservoirs carry limitations related to the maximum volume of stored water. Although the water sources are considered infinite (in a certain time scale), the volume of a tank isn't. It may overflow if the demand is lower than the collection or drain if the demand is higher.

1.2.2. Problems in the water sector

Nowadays a typical water reservoir is managed considering the immediate demand of the network. When the water level hits a predetermined minimum, its refilling system – pumps and valves – is activated. That system is only deactivated when the predetermined maximum level is reached, not taking in consideration any factors related with periodic electricity tariffs, making the system energetically and financially inefficient. There's margin to improve the operation efficiency of the pumps at the storage level keeping in mind that the water existent in the tanks is always enough to satisfy the network demand. Nonetheless, finding the optimum operation scheme of the equipment taking in account the electricity tariff and water demand is a problem with a complex solution [3].

1.2.3. Solution

The efficiency of equipment operation can be improved by taking advantage of the electricity tariff, favoring its operation when the electricity is cheaper and avoiding the more expensive periods. By predicting the water demand at short-term (24h-48h) the pumping schedule can be planned to operate at the cheapest periods, always guaranteeing the existence of minimum water in the tank. A decision support system is useful in this process, and it should be composed of two distinct modules. The first deals with forecasting the water demand, using the available and relevant data. The second bases its calculations in the forecasts made and the network's information and tries to come up with the optimal operation schedule.

In addition, the water company should negotiate with the power company a tariff schedule that minimizes the cost of this new operation schedule, keeping in mind the identified demand patterns. Forecasting the water demand of the network at each moment is necessary for the efficient management of the equipment and for an effective renegotiation of the tariffs.

1.3. Importance of forecasting

For a long time mankind has tried to predict the phenomena that rule the world and universe. Prediction includes the study of the phenomenon, finding a pattern in it and formulating a trend. There is a direct relationship between prediction and the existence of patterns.

The first steps in predicting the natural world were taken when men realized the pattern in the movement of the sun and the moon, unveiling the secrets of seasons and allowing for advances in agriculture techniques. With the passing of time, men also started to perceive patterns in the climate, though with low scientific basis and reduced success rates compared to those achieved today. Today it is possible to make precise short-term weather forecasts, but that precision starts to fail at longer term forecasts. The chaos theory states that in dynamic systems, the slightest alteration in the initial conditions may result in a completely different outcome.

Forecasting methods are also very important in the financial markets, where the analysts goal is to predict economic global crises, companies' ruin or where and when to invest their money. In marketing, accurate predictions of the sales of a product and the market's reaction might mean the difference between a very successful product and a major flop. In the electricity supply market, the provider must know the consumption pattern of its network. Because this type of network has considerable low inertia, the production in the plants must be adapted to satisfy the network demand at every moment. These kinds of forecasts are only possible due to today's innovative mathematical techniques and computational processing power.

Water demand forecast obeys to some specifications, such as the inertia of the system, seasonality, type and number of clients, and the external factors that affect the consumption. Short-term (daily) forecasting has significant importance in the previously described problem, allowing for an efficient management of the water existent in the storage tanks and of the equipment associated with it. Long-term (annual) forecasting is essential in the water network design phase. The knowledge of the consumption pattern of the region allows the project planner to better dimension the network's equipment. It also allows to analyze if the water sources have enough water to supply the network during the total period of the investment. Furthermore, it allows governs to assure the access of the entire population to water.

1.4. Difficulties and challenges in forecasting

The resolution of the problem described in 1.2.2 is achieved by finding a forecasting technique that is precise, effective and fast.

The use of numeric algorithms assures a high speed due to the capacity of today's computers. In current times there are resources available that allow anyone to run, in its own computer, algorithms capable of finding patterns in the data and, from those patterns, make a prediction. Still, the available resources are developed to solve generic problems, and are not adequate to each specific problem.

In order to guarantee the precision of the results, one must study the different methodologies used in the resolution of forecasting problems and understand the mathematical background and its numerical implementations. It is also necessary to study the behavior of the water networks, namely which factors influence the demand and the requisites of the data, both in short and long terms. Not taking into consideration an aspect may result in an inaccurate perception of the phenomenon. On the other hand, taking into consideration an irrelevant factor may result in a slow method with adulterated results.

1.5. Objectives

The main goal of this work is to find a practical solution that increases the efficiency of the operation of water supply systems, particularly of the equipment that feeds the reservoir tanks in the water supply systems, taking into consideration the price of the energy used. For that purpose, this work has the goal of contributing with the knowledge of the future consumption patterns, i.e. the forecast module described in 1.2.3. Artificial intelligence – machine learning – techniques are used to forecast the network's water demand at short-term. The results are validated using benchmarks and real data from two Portuguese water utilities.

2. State of the art review

2.1. Forecasting and predicting

In the past half century, as a consequence of the technological developments that have been happening during these years, there has been a search for new sources of knowledge. Until then, the only true source of knowledge was the human brain, which has amazing and uncovered capacities. However, at the time, the humanity is starting to teach machines to learn. These machines can do a vast range of tedious tasks, often more accurate and quicker than men. Current artificial intelligence techniques are good at recognizing complex patterns and tendencies, provided the correct rules for each type of task [4]. As a consequence of good pattern recognition, it is easy for the machines to predict the outcome of certain existing conditions. As a matter of fact, more than simply predicting that some phenomenon is expected to occur, today's techniques are very precise at forecasting the period and the magnitude of the phenomenon [4].

2.2. Machine learning

Artificial intelligence is a field of knowledge dedicated to develop ways to make machines and computers mimic human intelligence and behavior. By following a set of instructions for each different input the machines return an output that can be used to produce a decision (by men or machines). A subset of artificial intelligence known as machine learning goes further in this human brain mimicking and learning. A more abstract set of instructions is given to the machines, allowing them to adapt the outcome according to the objective function.

In machine learning, three main types of problems arise: supervised learning, unsupervised learning [5], and reinforced learning [4]. The first deals with datasets composed of inputs and outputs. For any paired input-output the machine must figure out how they relate to each another, allowing it to later estimate the probable output for a new untrained input. The second deals with problems in which the datasets used in the training process are composed only by inputs. The machine's task is to find the common features between each example and categorize them. As for the latter, it consists in a group of problems with no dataset. The computer generates its own dataset by running examples and evaluating the results.

2.2.1. Linear and logistic regressions

Linear and logistic regression is a group of algorithms that aims at finding a function that fits the training data available [5]. When more than one factor is thought to affect the outcome of a certain phenomenon, the influence of each factor is defined using weights. The phenomenon being studied is therefore described as a sum of smaller sub-phenomena. Finding the weights of each factor can be achieved using the gradient descent method [4]. The weight vector $\mathbf{w}^{t+1}$ in the iteration $t+1$ is found by

$$\mathbf{w}_j^{t+1}=\mathbf{w}_j^t-\alpha\cdot\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}_j} \tag{1}$$

where α is the learning rate and $J(\mathbf{w})$ is the cost function. The final linear regression function is given by

$$\hat{\mathbf{y}} = \sum_{j=0}^{n} \mathbf{w}_j \cdot \mathbf{x}_j \quad (2)$$

where $\hat{\mathbf{y}}$ and $\mathbf{x}$ are the regression's result and input data respectively.

Polynomial functions are a case of linear multivariable regressions where each power of the polynomial is equivalent to a different variable.

Logistic regression can be used in binary classification problems which outputs the probability of any given set of inputs (continuous or discrete) being classified as A. The probability of *x* being of class B is of course 1-P(A). It is commonly accepted by the engineer that when P(A)>0.5 the input is classified as A, and when P(A)<0.5 it is classified as B, leaving the case where P(A)=0.5 at a pseudo-random criterion. In some problems, it is considered as A and in others it is B, but it must be always the same inside the same problem. Now, to guarantee that the hypothesis only outputs values contained in the interval [0;1] it is used the sigmoid function, which output is greater than 0.5 above 0 and less than 0.5 when below 0. The goal of the logistic regression is to find the vector w that minimizes the misclassified instances.

Using logistic regression for more than two classes is done by finding a vector $\mathbf{w}_j$ that correctly categorizes each class via the aforementioned binary classification logistic regression. Finally, a new instance shall be considered as belonging to the class that gives the higher probability for its inputs.

2.2.2. Support vector machines (SVM)

The function of the previous subsection, although it can be multidimensional, will always have a linear form ($a_0+a_1x_1+\ldots+a_nx_n$). Support vector machines are classification algorithms that overcome this limitation by applying a non-linear transformation to the input [5]. Therefore, the support vector machine transforms the space where the two classes are only separable by a non-straight line into a new space where it is now possible to separate the classes using a straight line, also called a hyperplane for higher dimension problems. The desired transformation function is called kernel, and it takes a *n*-dimensional input and gives a (*n*+1)-dimensional output, where the two classes will presumably be linearly separable. Several types of kernel function can be used, such as circular, spherical, linear, polynomial or hyperbolic, just to name a few.

Multiple class SVM can be achieved by a list of adapted methods, either by finding a new single objective function or by running a binary classification SVM for each identified class using the remaining classes as negative examples (one-versus-the-rest) [5].

For regression problems, Support Vector Regression (SVR) can be used. The idea is similar to that of SVM, using a kernel function to transform a non-linear into a linear dataset, where the equivalent of a maximum margin hyperplane is calculated.

2.2.3. Artificial neural networks

Inspired on the biological neural networks, these networks process the information through a series of perceptrons that, because of their interconnections, will give a certain importance to different parts of the input information [4], [5]. From the simpler to the more complex, they all are made of (i) similar smaller units (perceptrons) that behave the same way as the others in the

same network and (ii) connections (synapses) that define how the perceptrons interact with each other. Neural networks can have multiple inputs and outputs, and can be applied in either classification or regression problems.

In the smaller scale (input-synapse-perceptron-output) the synapses connect the inputs to the perceptron, multiplying them by a weight $\mathbf{w}_i^k$, where $\mathbf{w}^k$ is the set of weights in the layer k. The perceptron can be described as a small machine that processes the information it is given. It takes the information given by each synapse connected to it, $\mathbf{x}_i\mathbf{w}_i$, and proceeds to their sum. To the result, it is applied an activation function, introducing nonlinearity, and the result of this operation is the output of the perceptron. In each layer of the network, all inputs must be connected to all the perceptrons. If a certain input is not relevant to the calculations, the algorithm will find a small weight for that synapse.

The backpropagation is the most commonly used learning algorithm, and it calculates each weight based on the difference between the output of each iteration and the target value observed. Other learning algorithms are described in [6] and [7].

2.2.4. Instance based learning

Unlike other methods, the instance-based method is based on storing training examples [4]. When evaluating a new instance, the method compares its information with the one of the examples stored in its memory, outputting a value close to the most similar instances studied. Because it doesn't have a global target function, this type of algorithm needs to evaluate each new instance, making it a slower solution if multiple instances are given at short intervals. For the same reason the algorithm needs no initial training other than memorization, making it easier to implement. Due to its implementation, the target function is substituted by simpler local target functions.

Instance-based algorithms can have multiple forms, such as *k*-nearest neighbor, locally weighted regression or radial basis function networks, being applicable to classification and regression problems.

2.2.5. Clustering

In unsupervised learning, a common problem is to identify and correctly classify a certain number of classes present in the data [5]. If *k* classes are thought to exist, a clustering algorithm will randomly allocate *k* points as cluster centers. Then each point of the data is compared to the existing cluster centers and is assigned to the one that is more similar to itself (i.e. smaller distance). For the next iteration, the new cluster centers are re-calculated as the average of all the points which were assigned to them. A variation of this algorithm calculates the new cluster centers when each point is assigned to any of them.

2.3. Creating the model

Although the best model is generally thought as the one that presents smaller differences between the forecast and the observation, that isn't always the case. Often the datasets used for the training of the algorithm contain measurement errors, noise, or random unpredictable occurrences. On the WSS, leakages, sporadic events or urban fires are some examples. Adjusting the model to fit these events will result in forecast failure. The sample error of an over fit model is smaller than a more general model, but the true error tends to get smaller on a

generic model. To avoid overfitting, one can use strategies such as early stopping the learning process or using separate sets of data for training and testing [4].

The methods used to predict water consumption usually consider a previous period of about two years. Holidays certainly have a high impact on the water demand of the network, but a two-year registry designed to avoid overfitting by simply eliminating outliers will fail to predict those events. The experience and sensibility of the engineer are crucial when designing the model.

2.4. Evaluating the model

Even though the main goal of machine learning algorithms is similar – mimic a real system – there are multiple evaluation procedures. While some methods are better at finding the overall pattern, ignoring occasional peaks, others can give a better understanding of sporadic events.

As stated before, when measuring the performance of a statistical experiment such as a demand forecast, there are two major dimensions: (i) how it describes the general tendency of the phenomenon, and (ii) how it behaves when it encounters possible random outliers and noise. When evaluating the general performance, a possible solution is to calculate the difference between each estimation (forecast, $\hat{\mathbf{x}}$) and the actual occurrence (observation, $\mathbf{x}$), i.e. the estimation error

$$\mathbf{e}_t=\mathbf{x}_t-\hat{\mathbf{x}}_t \tag{3}$$

Knowing the errors associated with each pair observation-prediction, the average error is given by the sum of each error, divided by the number of errors. Note that the average error doesn't reflect the real precision of the experiment, because its value can be zero even if the individual errors are non-zero but with positive and negative counterparts. However, it gives an idea about whether the forecasts are above or under the objective.

To avoid the problem stated above, the modulus function or the square function can be used, resulting in an average error always equal or bigger than the smallest individual error. Because of its formulation, root mean square error (RMSE) gives a higher importance to the higher errors calculated, thus being a better meter when doubling the error more than doubles the damage.

The difference between each error and the average error can also be used to analyze the estimates, by observing whether the estimates are close to their corresponding observations or not. Smaller deviations mean better forecasts, but also mean the forecast fits better to noise and outliers, which is undesired.

The Coefficient of Determination R^2 is a measure of how the difference between the observations and the forecasts relates to the difference between the observations and their average. It can be interpreted as the likelihood that new values are going to be correctly predicted. For the vector of observations $\mathbf{x}$ and the vector of forecasts $\hat{\mathbf{x}}$ the Coefficient of Determination is given by

$$R^2=1-\frac{\sum_{i=0}^{n_{\text{samples}}-1}(\mathbf{x}-\hat{\mathbf{x}})^2}{\sum_{i=0}^{n_{\text{samples}}-1}(\mathbf{x}-\bar{x})^2} \tag{4}$$

where $\bar{x}$ is the mean of **x** [8].

2.5. Validating the model

A model is not fully implemented as soon as it accomplishes the requisites of error and deviation proposed. Another step is necessary: validation. The developer must make sure its model has an acceptable behavior not only when applied to the dataset used in training, but mainly in the new data, which it was designed to work with. This validation process can be done by different approaches. The first one is to assess if the model is flexible and transversal by testing it in different datasets, often referring to different phenomena and with value ranges different than the problem being studied (benchmarks) [9]. Researchers and developers frequently use the same benchmarks, for reasons such as availability and ease of comparison of results. The second validation method, known as cross validation, consists in dividing the dataset in *k* equally sized folds, reserving one of them for testing purposes, while the other *k*-1 folds are used for training [4] and [5]. The model must be tested *k* times, each of them using the *k*-th fold as testing set. Regular results across the *k* testing sets are a better guarantee that the model is not overfitting. Another possible method to assess the robustness of the developed method is to use a mathematically generated dataset. By using a function that is well defined it is possible to compare the forecast values with the expected values given by calculating the function values in the same domain as that of the forecast.

2.6. Machine learning in water demand forecasting

The works presented in [10], [11] and [12] analyze the use of water consumption forecasting to improve the operation of WSSs. They concluded that, independent of the method used, forecasting the demand and deriving the operation schedule from the results can mean a cost reduction ranging between 18% and 55%. Although a cost reduction higher than 18% is not always guaranteed, these works show that a poor forecast is better than none at all, provided the decision maker takes it into consideration.

Recent studies such as [13], [14], [15] and [16] made their forecasts using a hybrid method which is decomposed in two steps: in the first the data is analyzed as a whole and the different patterns (clusters) are identified – unsupervised learning. In the second a different algorithm is applied at each identified cluster to produce reliable predictions.

In their paper, Candelieri et al. [13] describe a method to forecast water demand in the city of Milan, Italy. Their method is also divided in two steps: (i) identifying patterns in the water consumption data and (ii) predicting the water demand of the network for the next 24-*t* hours based on the first *t* hours of any given day. To identify the patterns in the data, Dynamic Time Warping (DTW) techniques were used on all the time series calculated for each 24-hour division of the data. They found 6 distinct clusters: three relative to periods of year (“Spring-Summer”, “Fall-Winter” and “Summer-break”), combined with two referring to types of day (“working-days” and “holidays-weekends”). The second step deals with the forecasting of the water demand for any given day, based on the consumption observed in the first hours of that day. Comparing the measured consumption of the first *t* hours of the day with the data contained in the identified clusters, and using a series of SVR models previously trained, the output is the predicted water demand for the remaining hours of the day. There is a different SVR for each combination of cluster-hour of the day.

Parallel studies [17], [18], [19], [20] and [21] have been made with the purpose of testing different forecasting methods and comparing their accuracy. Generically, it is shown that machine learning techniques (e.g. SVR and ANN) have higher accuracy than non-learning approaches (e.g. time series). Herrera et al. [17] tested a group of forecasting algorithms on a dataset corresponding to a city in south-eastern Spain. They concluded that the methods can be ranked considering their accuracy as follows: heuristic model, ANN, random forest, projection pursuit regression, multi-variative adaptive regression splines, and support vector regression. Furthermore, comparison tests have also been made with different configurations of each method.

De Lima et al. [18] studied three forecasting methods: Exponential Smoothing (ES), Seasonal Autoregressive Integrated Moving Average (SARIMA) and ANN. Additionally, they applied those methods in data from 10 cities in Paraná, Brazil. Then, 14 combinations of the results were evaluated to assess the best. They concluded that more complex methods don't mean better results, since ES was found to be the best method in 5 cities, SARIMA in 4 cities and ANN in just one city. The best model in each city showed values of MAPE less than 4%.

Tiwari et al. [19] compared 6 models, consisting of 2 methods - Extreme Learning Machine (ELM – a derivation of Neural Networks with a single hidden layer where the weights of the neurons are randomly assigned) and ANN - with 3 different implementations - traditional, wavelet analysis and bootstrap. These methods used 3 years of water demand and climate registries of a network in Calgari, Alberta, Canada. The ELM and ANN methods achieved similar results, and suffered no significant improvement when using the bootstrap method. However, significant improvements were observed when the wavelet analysis was applied to both ELM and ANN.

Peña-Guzman et al. [20] applied Support Vector Machines to a real network located in Bogotá, Colombia, and used previously observed water consumption, number of users and the value billed for monthly consumption data in their forecasts. They analyzed 6 residential sub-networks, 1 commercial sub-network and 1 industrial sub-network. Except for one residential sub-network, all the others showed a RMSE<2% and Coefficient of Determination R^2>0.9. Moreover, they found that the LS-SVM used achieved better performance than the Feedforward Neural Network Backpropagation FNN-BP tested for comparison.

Ghiassi et al. [21] used three machine learning methods (Dynamic Artificial Neural Network (DANN), Focused Time-delay Neural Network and *k*-Nearest Neighbors) to forecast the urban water demand in Tehran, Iran, for three time horizons: 4 weeks, 6 months and 2 years, using respectively daily, weekly and monthly time steps. Their methods used the daily water production and monthly water consumption data between March 2003 and April 2009 provided by the Tehran Water & Wastewater Company. They tested two methods for the daily forecasts, where they studied the impact of partitioning the weekdays into weekends and non-weekends. They found that the best results were achieved when this partitioning was not considered. For the weekly and monthly forecasts, they evaluated whether using the daily data for the forecast and then integrating to the time is better than using the weekly/monthly data followed by the forecast. The results were improved by integrating after the forecast. Additionally, they also tested the monthly forecast taking into consideration seasonality (high and low seasons), and observed a positive impact of this decision. Generically, the three developed methods were considered to provide good results in the three time scales, with a slightly better performance of the DANN.

Brentan et al. [22] developed a hybrid method in which they make a base prediction using SVR, followed by the application of an Adaptive Fourier Series (AFS) to improve the previous

forecast. This method was validated using the dataset of a water utility in Franca, Brazil. They used previously observed consumption and weather data (rain, temperature, humidity and wind velocity) in the process. They also considered the yearly seasonality (on a monthly basis) and the difference between weekends, non-weekends and holidays. The comparison between the developed hybrid method and the basic SVR method proved that applying the AFS resulted in a much better forecast: RMSE from 4.767 to 1.318 L/s, MAE from 12.91% to 3.45% and Coefficient of Determination R^2 from 0.745 to 0.974.

Shabani et al. [23] used Phase Space Reconstruction to derive the proper lag time (found to be three months) to be used in their Genetic Expression Programming (GEP) method, which aimed at predicting the average water demand for the entire next month. In the dataset considered, they found a high correlation between the water demand and the temperature and hotel occupancy, which seems to reflect the seasonality on the case being studied. The population of the city and the rainfall didn't show a high correlation with the water demand forecast. The GEM with best performance was then compared with SVR with different kernel functions - radial, linear and polynomial – and the polynomial was found to be the best, not only amongst the SVM, but also amongst all the methods evaluated. The results were validated using data referent to City of Kelowna, British Columbia, Canada.

In their work, Moutadid and Adamowski [24] used combinations of water demand data (1999-2010), maximum daily temperature and daily total precipitation referent to the city of Montreal, Canada, and forecast the water demand with 1 and 3 days of lead time. ANN, SVR, ELM and the traditional Multiple Linear Regression models were developed, being the ELM the one which presented the best performance independent of the lead time. They also observed that an increase in the lead time means a worse forecasting, even though this decline is considered by the authors as not drastic.

Haque et al. [25] showed an innovative regression method named Independent Component Regression (ICR) and applied it to medium-term (monthly) water demand forecast in Aquidauano, Brazil. For comparison, they also calculated forecasts using multiple linear regression and principal component regression. They used monthly history data of maximum temperature, relative humidity, number of water customers and water consumption. The results showed that even though the R^2 of the ICR method was lower, the other evaluation parameters proved its high performance. An overestimation tendency of the ICR method was observed.

Galiano and Claveria [26] applied regression trees as a water demand forecasting technique. The model used socio-demographic data such as age of the population, cadastral value and size of the buildings and derivatives of these. In total, 15 variables were used as input vector in the training process. The domestic water consumption history was used as the output target vector in the training process. They evaluated the RMSE when using n variables with more impact in the forecasts and observed that when using only 1 variable (household size [inhab./household]) the RMSE=26.91 L/y. and using the 15 input variables the RMSE=18.89 L/y. As a consequence of using the n most important input variables, they also observed that the last input variables used had little impact in the forecasts. The RMSE calculated when using only the 6-most important variables was RMSE=18.96 L/y. Considering more variables has an insignificant improvement in the results, with a higher computational cost. The tests were performed using data relative to a WSS in Sevilla, Spain.

Melios et al. [27] developed an Artificial Neural-Fuzzy Inference System (ANFIS) to forecast the daily water demand in the Greek turistic island of Skiathos. They used daily water pumping history, daily mean and high temperature, daily precipitation, daily wind speed and monthly arrivals by air and sea regarding a 2-year period (2011-2012) for training and 2013 for

testing. From the 32 networks evaluated, the best showed the following results: R^2=0.916; ME=82.86 m^3; RMSE=192.99 m^3; MAE=151.23 m^3; MAPE=8.1%.

In their work, Suh and Ham [28] used BP-ANN to forecast the water demand in buildings of 4 cities in South Korea. Using climate, geographic, and morphologic input variables and average monthly water consumption as output in the training process, they could predict the monthly water consumption with a MAPE=19.6% and RMSE=98.11 m^3/y.

Seo et al. [29] used three wavelet decomposition methods to assess their ability to predict the water level of a dam. They used the ANN and ANFIS methods and their decomposed variants WANN and WANFIS and validated the results using real daily water level data for the Andong Dam in South Korea. The results showed that the ANFIS methods are generally better than the ANN. Furthermore, the application of the wavelet decomposition resulted in significantly better results and the best method is identified as WANFIS7-sym10 – input set 7 with Symmlet-10 wavelet decomposition.

Adamowski and Karapataki [30] observed that for peak urban water demand forecasting, and in the two datasets used (networks in Nicosia, Cyprus), the accuracy of the learning algorithms can be ranked as follows: multiple linear regression, resilient back-propagation ANN, conjugate gradient Powell-Beale ANN, Levenberg-Marquadt ANN.

Some studies have been made on the influence of the weather as input data [26], [30]. They all concluded that the weather influences the water demand, mainly on the domestic and agriculture levels. On the other hand, opposing conclusions have been presented concerning the impact on the forecast of either the quantity of rain or the occurrence of rain [30]. Although the use of weather data as input was shown as a performance enhancer of the methods used, it is also well understood by the community that the difficulties of implementation of such methods do not always pay off the additional effort. Bakker et al. [15] developed a method that takes into consideration the weather effect, even though they do not use any weather data input. In this study, they also showed that a shorter time interval helps modeling critical times of the day (early morning), but results in a smaller overall accuracy.

Forecasting techniques have also been used to predict malfunctions of the equipment and locate leakages. Candelieri et al. [16], used spectral clustering and SVR techniques with the aid of a simulated water supply network. They achieved a reliability of 98 % for pressure and flow variables and leak locations. When applied to real cases (Milan and Timisoara, Italy), this technique achieved a reliability larger than 90%.

3. Developing a machine learning water demand forecasting model

According to previous studies made on the field it is not expected that a particular machine learning model is found to be the most adequate for every water demand forecasting problem. However, it is expected that some methods present better predictions than others, for different datasets. Developing a flexible, transversal and accurate algorithm involves studying a variety of methods applied to multiple databases.

3.1. Choosing the features

The selection of the right input features[*] for the training stage can mean the difference between a poor and an excellent forecast. The scientific community has made several studies concerning the most adequate features for water demand forecast and it can be generically concluded that, besides the historic data, the weather and seasonality have the strongest impact on the results.

In this context, seasonality must be considered at several levels and is to some extent correlated with the weather data. Yearly seasonality means it is different to predict a winter day or a summer day, weekly seasonality accounts for the fact that a typical Monday is different from a typical Tuesday or Sunday and daily seasonality expresses the hours that typically have higher and lower demand. The sporadic seasonal events, such as Christmas, Easter or even a major sports event can also be considered. The seasonality is implemented in the algorithm in two different ways:

- the periodicity of the data directly affects the number of machines used in the forecast and consequentially the amount of data used in the training of each of them. For example, considering a periodicity of 24 hours means the algorithm will train 24 machines and make forecasts for a 24-hour interval. The different periodicities to consider have different applications and may have implications on the accuracy of the forecasts.

- the brute-force clustering, where the algorithm is told that each day of the week corresponds to a different cluster and during training the machines will only use data of the same cluster. By using this approach[†], the developer assumes that different days of the week have different typical behaviors and therefore must be predicted based on solely on those of the same pattern. Additionally, sporadic events may be considered as one of these clusters. In this paper, the two clusters considered are (i) weekday, including the days between Monday and Friday and (ii) weekend, including Saturdays, Sundays and the Portuguese official holidays.

As for weather features the works made on the subject concluded that temperature, rain amount and rain occurrence have the largest impact on the training stage. For this reason, when available, these registries are considered as features for the forecast. Nonetheless, models[‡] with no weather features are also tested. Regarding this matter, it is considered that the water demand forecast depends on the previous water consumption observations in pair with the predicted weather conditions. For example, the forecast for the hour 14 of tomorrow depends on the water consumption observed at the hour 14 of today and the temperature forecast for hour 14 of tomorrow. The weather variables are not predicted, as they are usually available on external sources, and are not the object of this work.

---

[*] In this context, each feature represents an input variable that affects the outcome of the forecast. Some examples are the water demand history, temperature or rain occurrence. Note that the temperature observed at the instant t and the temperature observed at the instant *t*-1 can be two features both referents to the same instant *t*.

[†] An approach is the group of features applied to each model. One approach might be using 70 water demand features, and another approach might be using 14 water demand and 1 temperature features.

[‡] A model is a well-defined forecasting method configured and trained.

The last consideration is related to the amount of data that is used in the process. In this work, the aim is to use as much available data as possible. If two years of records are available, it is not advised to use any less than those two years of data. However, the water consumption observed two years ago has not a direct impact on tomorrow's demand. All the data must be used during the training, but only a parcel of that has a direct influence in each step of the process and, consequentially, in the forecasting. The parcel of data used is updated in each training step, but maintains the same size. This implies that the water consumption for the day D_{+1} is predicted considering the features registered in the days D_{-n} to D_{-1}, and each of those days is a sample[§]. Additionally, the number of weather features considered must also be tested, and it is not mandatory that it equals the number of demand observations considered. In other words, tests are made considering only the weather forecast for one day or weather forecasts for 14 days (starting from the day to predict, moving to the past).

3.2. Forecasting techniques

The water demand forecast can be seen as a regression problem, and many machine learning methods have been studied in the past. Based on these two statements, the following methods arise as candidates for solving the problem: Artificial Neural Networks, Random Forest Regression, k-Nearest Neighbors, and Support Vector Regression.

3.2.1. Artificial Neural Networks

The design of a neural network can be divided in 2 steps. The first deals with the network's morphology, i.e. the number of layers and the number of neurons in each of them, as well as the activation function applied at each neuron. A collection of neural network architectures and its description is presented by van Veen [31]. There, the Feed Forward Neural Networks (FFNN) are described as simple and practical, and are used in this work for ease of implementation. FFNN are trained neural networks in which the new information travels in from the input to the output. Svozil et al. [32] refer some advantages of this method, namely its learning process being autonomous from the user, its application in non-linear problems, the resistance to noise and the fact that each input set generates a trained model fully adapted and adequate to that same problem, preserving the idea that no two problems should have the same solution. They refer the slow convergence and unpredictability associated with difficult interpretation of results associated with this method as the major disadvantages.

Using a non-linear activation function means that the output of a neuron cannot be expressed as a linear combination of its inputs. Without this step, every neuron would output a combination of its inputs, and in the end the solution would itself be a combination of the initial inputs. A neural network without non-linear activation functions is a neural network that can be simplified to a single layer. For simple problems that can be described with a linear model the generic need for activation function is fulfilled using the identity function

---

[§] In this context, a sample represents a moment of observation, each consisting of the features used by the machine (in training and predicting). Each sample is a vector of the features' values observed at any given moment.

$$f(x)=x. \tag{5}$$

The rectifier function is defined by

$$f(x)=\begin{cases} 0, & x<0 \\ x, & x\geq 0 \end{cases} \tag{6}$$

and assures the non-linearity with low computational effort. It assures that the output of the perceptron has a positive infinite range. Somewhat similar to each other, the logistic and the hyperbolic tangent functions, respectively

$$f(x)=\frac{1}{1+e^{-x}} \tag{7}$$

and

$$f(x)=\tanh(x), \tag{8}$$

assure that very large positive or negative numbers are approximated to the same value (1 if x is a large positive and 0 (logistic) or -1 (tanh) is x is a large negative). It also ensures that around x=0, f(x) is approximately a linear function.

The second consideration relates to the training process occurring in the neural network, embracing the learning algorithm and learning rate. Gradient descent is an optimization method often used in machine learning problems, where it is used to find a minimum of the cost function f($\boldsymbol{\theta}$). Until convergence it iteratively calculates

$$\boldsymbol{\theta}^{i}=\boldsymbol{\theta}^{i-1}-\alpha\nabla\left(f\left(\boldsymbol{\theta}^{i-1}\right)\right), \tag{9}$$

where $\boldsymbol{\theta}^{i}$ are the model fitting parameters found at the i-th iteration, α is the learning rate and ∇ represents the gradient operator. The use of a constant learning rate carries two possible unwanted outcomes. Too small and it converges unnecessarily slowly, too large and it may fail to converge. Choosing the learning rate is frequently done by trial and error. Alternatively, one can use an adaptive learning rate as an attempt to avoid these issues. The Adaptive Moment Estimation (Adam) [6] method is an adaptation to the gradient descent, as it recalculates the learning rate at each iteration, applying exponentially decaying average of previously observed gradients of first and second order. Another possible way to bypass the disadvantages of stochastic gradient descent (SGD) is to use a second order optimization method such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm described in [7], where the vector $\boldsymbol{\theta}^{i}$ is calculated based on the hessian of f($\boldsymbol{\theta}$). In each step, the learning rate is updated such as its value assures that f($\boldsymbol{\theta}^{i+1}$)/f($\boldsymbol{\theta}^{i}$) is smaller than a stipulated value. The BFGS algorithm generically guarantees that fewer iterations are required, but because of the heavy calculations associated with the hessian matrix, those take more computational time than SGD. Limited Memory BFGS (LBFGS) is an implementation of BFGS designed to overcome this issue [33].

3.2.2. Random Forest Regression

A Regression Tree is a method that was first developed to solve classification problems. It was later adapted to regression problems, although it does not provide a continuous output space. When given a new sample, the model will proceed to a series of comparisons starting from the root of the tree and following the path that respects all comparisons made. When a node where no splits exist is reached the process finishes and the output is calculated according to that node [5].

The method starts by mapping each sample into a *N*-dimensional space, where *N* is the number of features thought to influence the phenomenon. Then, along each direction, the data is split in a way that minimizes the Mean Square Error in each of the two newly generated regions. This process stops if certain error criterion is satisfied or when maximum depth of tree is reached. The deeper the tree the better the fitting, thus opening the possibility for overfitting. The tree is built in a way that by navigating through it, all training samples are correctly classified.

A random forest is a collection of random trees, being the output of the model the average of the individual outputs of each tree. In this case, each individual tree takes a portion of the data, resulting in slightly different trees. This technique is used as a way to minimize noise-induced errors.

3.2.3. K-Nearest Neighbors

The *k*-Nearest Neighbors (KNN) algorithm makes a forecast by making a pondered average of the *k* vectors most similar to the input vector currently being assessed. Given the input vector x^i it is calculated the distance between it and each vector present in memory [5]. The Minkowski distance [34] can be used to calculate the Manhattan distance and the Euclidian distance. The distance function is important in this method not only to find the nearest vectors but also to find the weights later assigned to them. The output of the algorithm is a weighted average of the *k* vectors found, where the weights are usually proportional to the distance to the input vector. Alternatively, using uniform weights means that each of the *k* vectors is assumed to have an equal impact on the outcome. This method is simple and fast, and for this reason it is usually one of the first methods tested when studying a machine learning problem. However, when accuracy has a bigger importance than simplicity it is often surpassed by other methods.

3.2.4. Support Vector Regression

The goal of a support vector machine is to find a function that separates the two classes in the classification problem being studied. For regression problems, the Support Vector Regression (SVR) method can be used, where the goal is to find a function f that better fits the training data. When a new vector **x** is used as input, the output is calculated by computing f(**x**). Immediately, a disadvantage over ANN arises. While other methods can have multiple outputs, SVR only allow single outputs.

Given a dataset it is possible that more than one hyperplane correctly classifies all data points, but the desired solution is the hyperplane that is equally distant from both classes, providing a better generalization, necessary for new data. After the hyperplane f(**x**) is found, a classification problem is tackled by computing f(**x**) followed by

$$\begin{cases} \mathbf{x} \in A, \text{ if } f(\mathbf{x}) > +1 \\ \mathbf{x} \in B, \text{ if } f(\mathbf{x}) < -1 \end{cases} \quad (10)$$

where A and B are the two classes. More generically, in a regression problem the output is simply given by $f(\mathbf{x})$ for each new instance $\mathbf{x}$ being evaluated.

3.3. Configuring the models

As shown by the literature, different techniques are better suited for different systems. For this reason it is not expected to find a solution that perfectly fits all datasets, neither to find the perfect solution for each methodology. However, it is expected to find which model configurations being tested present the best results. The present strategy contains a set of models varying only one parameter between them, in order to evaluate their influence. Only the parameters that are expected to have the largest impact in the definition of the model are considered.

A neural network is essentially defined by the shape of the network itself (number of neurons and number of layers), activation function and learning algorithm. 9 shapes of networks will be tested, being those combinations of 3 numbers of layers (2, 5, 8) and 3 numbers of neurons per layer (10, 25, 75). 3 activation functions will be studied: identity, logistic sigmoid and rectified linear unit. The learning algorithms to be tested are SGD, LBFGS and Adam.

The support vector regression machines are highly dependent of the kernel they use. Radial Based Function (RBF), linear and polynomial (second degree) are the kernels tested. Two values of tolerance (0.01 and 0.001) for stopping each learning iteration are also tested. The learning algorithm stops when that tolerance is satisfied.

The *k*-Nearest Neighbors method is strongly defined by the number of neighbors considered relevant to the calculations and by how the weights are calculated. Tests with 3 numbers of neighbors (2,5,8) and 2 weight functions (uniform and distance) are considered. The uniform weight assigns the same weight to the *k* neighbors considered, while the distance weight function gives a weight proportionally inverse to the Euclidean distance between each neighbor and the data.

The random forest method can be modeled by the number of trees in the forest and the minimal number of samples required at each split. With this in mind, 6 combinations of 3 numbers of trees (2, 8, 15) and 2 numbers of samples required to split (2, 8) are analyzed.

In total, 99 model configurations are tested, being 81 ANN, 6 SVR, 6 KNN and 6 RFR.

3.4. Implementing the models

The Scikit-learn 0.18.1 [35] library for Python 3 [36] allows the creation of machine learning models in a simple and efficient way. Other libraries from the SciPy [34] environment ease the data manipulation (NumPy and pandas) and the data visualization (Matplotlib). The overall algorithm is schematically represented in Figure 1. It starts by reading the data file. Then, it applies a filtering routine that eliminates outliers, missing values and normalizes the data (described in 5.1), and rearranges the data in a three-dimensional matrix. This rearrangement is done to overcome the limitation imposed by the Scikit-learn library in its fitting function, since it only allows two-dimensional matrixes as entries. Although other

shapes for this matrix would be possible, this presents an easier visualization of the data matrix. In the three-dimensional matrix represented in Figure 2 each horizontal plane represents an hour of the periodicity considered, where each line represents a sample (if periodicity is 24h a sample is 1 day, if periodicity is 168h a sample is a week) in the training data and each column represents a feature.

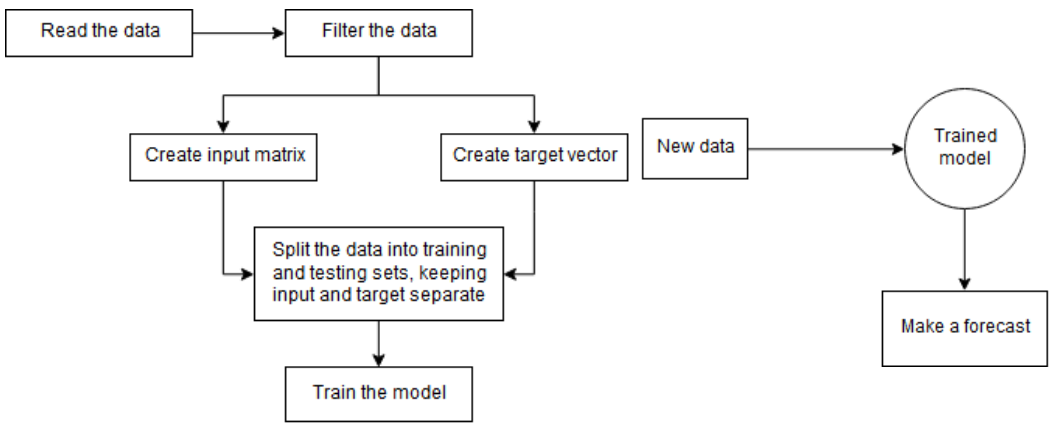

Figure 1 - Schematic representation of the developed algorithm. (a) Training the model and (b) forecasting procedure.

When fitting a model, Scikit aims at optimizing the RMSE between the observations and the estimations. Because this is the metric used by the program when developing a forecast, it must be the metric used when comparing results. Generically, a model is considered better than another if its RMSE is lower, although other metrics can also be useful in specific cases. The RMSE values are affected by the fact that the data is normalized, and for this reason it is presented as a dimensionless metric.

In addition to the input data matrix, a target matrix is created. This target matrix only has one column as features - the observed consumption. The algorithm then divides the available data into training and testing sets, reserving the last split for testing, using the *TimeSeriesSplit* (Scikit-learn) function with 32 folds (28 folds when periodicity=168h). Cross-validation is not used because the data is time sensitive and altering the order of the data would affect the training and the results. Then the program creates a number of machines equal to the height of the matrix (periodicity) and trains each of them using the two-dimensional matrix corresponding to the *T*-th plane of the bigger matrix as input data and its corresponding target vector in the training set. This way, the *T*-th machine predicts the *T*-th hour of the periodicity considered and stores its value in the *T*-th element of a one-dimensional "Forecast" vector. Finally, the program compares the forecast result (for the period being studied) with the testing data for the *D* days in the testing set. A graphical representation of the forecast and the testing data, as well as the evaluation results are stored in external files.

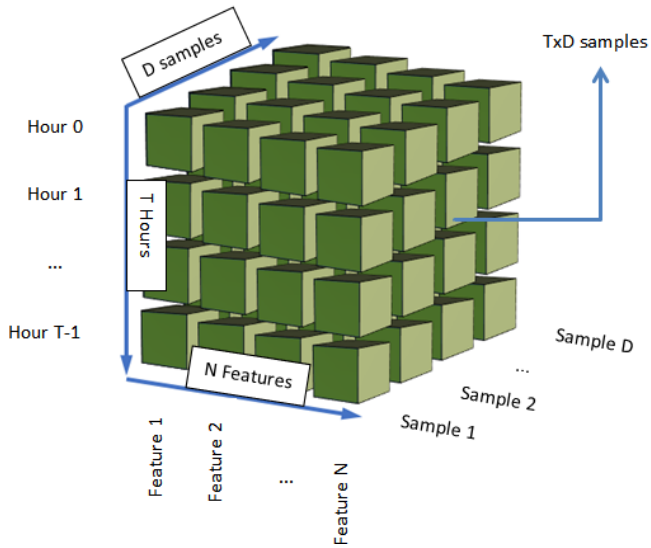

Figure 2 - Representation of the data matrix used in the machine learning models. Each sample has *N* features. Adapted from [37].

3.5. Parallel methodologies

In section 3.3, the advantages and disadvantages of each presented forecasting technique were presented. The advantages of each technique can be combined through a hybrid strategy of parallel methods, hopefully diluting the disadvantages. In this work, a weighted parallel strategy is suggested.

Considering that some methods tend to overestimate the demand and others tend to underestimate it, one can improve the forecast by calculating a weighted average of the various forecasts previously made (i.e. one for each model tested). The weight of each model must reflect the robustness of the forecast made. Attributing the weights to the desired models is not just a matter of evaluating the errors and assigning a higher weight to those with smaller errors, but to combine over and underestimations in a way that diminishes the error. A thorough analysis must be done to understand the models that over and underestimate the forecast and its magnitude.

The hereby proposed methodology includes several steps. The first step includes the forecasts made by the models from the previously discussed set. The performance of each model is considered as the average of the results found to each individual day in the testing set. In the next step, three models with positive Mean Error and three models with negative Mean Error are chosen. In each case, the chosen models are those which present the best RMSE, MAPE% and R^2. This methodology presents a new forecast, based solely on a combination of the forecasts previously made. Each of the selected models has an associated weight proportionally inverse to the absolute value of its Mean Error. Therefore, the models with lower errors have a higher impact on the forecast. For that, each weight $\mathbf{w}_i$ is the inverse of the Mean Error given by its corresponding model. The final forecast is given by

$$\mathbf{F}=\sum_{i=1}^{6}(\mathbf{F}_i\cdot\mathbf{w}_i)\cdot\frac{1}{\sum_{i=1}^{6}\mathbf{w}_i} \tag{11}$$

4. Validation of the machine learning algorithms developed

To effectively use the developed algorithm in predicting water consumption one must make sure the algorithm achieves its goal and with advantages compared to other algorithms. This process can be done by running the algorithm using known and/or predictable data, either real data often found in the community, or mathematically generated data. By comparing the results achieved by the developed algorithm with those presented by other algorithms one can assess the viability, applicability and quality of the forecasting program algorithm. The method considered for comparison is the Autoregressive Integrated Moving Average (ARIMA), recommended for time series problems.

4.1. Benchmarks tested

4.1.1. Periodic Function

Consider a hypothetical network with a daily water demand pattern that repeats itself infinitely. This pattern has a valley during the night, a peak in the morning and another in the evening. Moreover, a weekly and a monthly seasonalities are added, and then a random noise, thus giving for each day a slightly different pattern. The function used in this process is defined as

$$Q(t)=50+\frac{f_1(t)+f_2(t)+f_3(t)+f_4(t)+f_5(t)}{5}+\text{GAUSSIAN}(0,2) \tag{12}$$

where

$$f_1(t)=30\sin\left(3+\frac{2\pi}{12}t\right), \tag{13}$$

$$f_2(t)=30\sin\left(-3+\frac{2\pi}{24}t\right), \tag{14}$$

$$f_3(t)=30\sin\left(10+\frac{2\pi}{24}t\right), \tag{15}$$

$$f_4(t)=20\sin\left(-3+\frac{2\pi}{168}t\right) \text{ and} \tag{16}$$

$$f_5(t)=10\sin\left(0+\frac{2\pi}{744}t\right) \quad (17)$$

and GAUSSIAN(0,2) represents a Gaussian noise with mean 0 and standard deviation 2. The functions $f_1(t)$, $f_2(t)$ and $f_3(t)$ model a daily behavior since they have periods of 12h, 24h and 24h, respectively. The functions $f_4(t)$ and $f_5(t)$ represent a weekly and monthly tendencies, defined by their periods of 168h and 744h. The data observed in the last 2 days is shown in Figure 3.

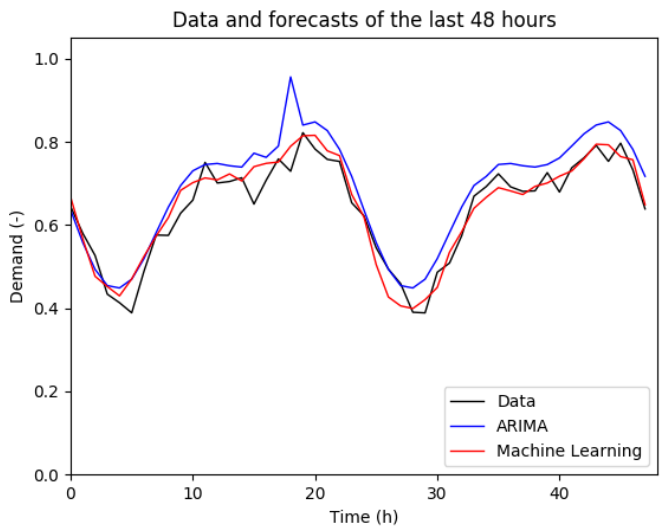

Figure 3 - Mathematically Generated Data Benchmark . Water consumption observed in the last 2 days. Forecasts given by the ARIMA and KNN(N=8 euclidian) models.

4.1.2. Cars Benchmark

This benchmark is based on a dataset used in three Artificial Neural Network & Computational Intelligence Forecasting Competitions, held between 2009 and 2010 by the Lancaster University Management School. The database consists of a collection of traffic data, including highways, subways, flights, shipping imports, and railways. The entire dataset is presented in 4 parts of 1735 instances plus 5 parts of 895 instances, but a quick analysis shows that these do not represent a pure sequence of data. For this reason, only 1 part with 1735 is used. This means that only 72 days are available to test the machine learning models presented. The amount of data available brings an extra difficulty, derived from the small number of iterations during training.

4.1.3. Air Quality Benchmark

The Air Quality Benchmark contains the data collected by an equipment that measured the quality of the air in regular intervals of 1 hour in an Italian city. In total, 9358 (389.91 days) measurements were registered. The data considered in the calculations was however reduced to

assure it has a length divisible by the periodicity considered in each test (9336 registries used for the periodicity of 24h). When predicting water demand, the method proposed considers a maximum of 2 types of features (past demand and a meteorological variable). For this reason, for the benchmark tests using the Air Quality dataset, only 2 out of the 14 types of features available were selected: True Hourly Averaged NOx concentration in ppb (reference analyzer) and Temperature in °C. The Average NOx concentration feature was chosen because its range is similar to that of a typical water demand in cubic meters. The data presents some values of -200, specifically used to avoid implementation errors associated with missing values. However, knowing that these are outliers, they are submitted to a filtering routine described in 5.1. For availability reasons this is also the only benchmark that considers meteorological features, bringing it closer to the real applications intended for the developed methodology and program.

4.2. Results

The evaluation of the developed models is done in two consecutive steps. First, each sample of the testing set data is evaluated, and second the average for each metric is calculated considering the entire testing set. The metrics evaluated are the Mean Error (ME), Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE%) and Coefficient of Determination (R^2), but only the RMSE (used by the Scikit libraries in the fitting process), MAPE% and R^2 are shown. To assess the performance of each methodology, the results for the best model in each family (RFR, KNN, SVR and ANN) considering both metrics are compared to those of ARIMA methodology.

4.2.1. Results for the Mathematically Generated Data

Using the Mathematically Generated Data, the performance of the 99 models designed was evaluated. Between the periodicity thought to rule the phenomenon and the past observations thought to influence the forecast, it is analyzed the influence of these two parameters, fixing one of them and varying the other. Fixing the periodicity at 24h, tests were made considering 3, 14 and 70 past observations. Defining the past observations at 14 samples, tests were made considering periodicities of 12, 24, 48 and 168 hours. The tests made without considering noise show perfect forecasts (RMSE=0, MAPE=0 and R^2=1) for the RFR, KNN and SVR methods in every approaches, except when using clustering (where the best model, ANN(identity(2x10) sgd), obtained RMSE=0.0241). When using Gaussian noise, the best models per approach are presented in Table 1. There, it is visible that the different approaches tested present similar results for all metrics shown, with the exception of the coefficient of determination when using 12-hour periods in the forecasts. This behavior can be explained by comparing two consecutive periods generated by the function. The consumption patterns observed in the first 12h and in the last 12h of any given day are obviously different. Using the data of the first half of the day to predict the second half is not advised, as often stated by the literature. The performance of the models improves when the number of features increases, but that (or its contrary) cannot be said about the time scale of the forecasts. In fact, the lowest RMSE is found when using 70 water demand features with a 24h periodicity. It is also observable that the KNN models are usually the best choice.

In Table 2, the best results overall of each family of models (RFR, SVR, KNN and ANN) and for ARIMA are presented, along with the best three models, considering the best approach found in Table 1. The results obtained by the best machine learning models in any method are

satisfactory, with their RMSE at least 32% lower than the one observed with the ARIMA. However, as proved by the results obtained by the worst model (a Neural Network using SGD as learning algorithm and the logistic activation function), the use of machine learning does not guarantee good results. Figure 3 presents the results obtained with the ARIMA and KNN(N=8 euclidian) models.

Table 1 – RMSE, MAPE% and R^2 of the best model found with each approach tested for the Mathematically Generated Data Benchmark.

Periodicity	Features	Model	RMSE (-)	MAPE (%)	R^2
24h	Demand (3)	ANN(identity(2x25) lbfgs)	0.0392	4.7678	0.9086
24h	Demand (14)	KNN(N=8 euclidian)	0.0304	3.6355	0.9417
24h	Demand (70)	KNN(N=8 euclidian)	0.0284	3.4653	0.9476
12h	Demand (14)	KNN(N=8 euclidian)	0.0323	3.9408	0.6677
48h	Demand (14)	KNN(N=8 uniform)	0.0315	3.8536	0.9411
168h	Demand (14)	KNN(N=8 euclidian)	0.0300	3.6170	0.9510
24h	Demand, using clustering (14)	ANN(identity(2x10) sgd)	0.0408	4.8732	0.8919

Table 2 – Models that achieved the (i) best RMSE per family, (ii) the overall best 3 RMSE and (iii) the overall worst RMSE, and (iv) the ARIMA results, applied to the Mathematically Generated Data Benchmark using 70 24h demand samples.

Model	RMSE (-)	MAPE (%)	R^2
RFR(N=8 n=2)	0.0305	3.7452	0.9400
KNN(N=8 euclidian)	0.0284	3.4653	0.9476
SVR(linear t=0.001)	0.0352	4.3049	0.9210
ANN(relu(5x25) lbfgs)	0.0306	3.7358	0.9392
KNN(N=8 euclidian)	0.0284	3.4653	0.9476
KNN(N=8 uniform)	0.0285	3.4647	0.9475
KNN(N=5 euclidian)	0.0290	3.4713	0.9452
ANN(logistic(5x10) sgd)	0.1319	17.8044	-59.4768
ARIMA	0.0523	6.7217	0.8149

4.2.2. Results for the Cars Benchmark

The dataset used by this benchmark has the particularity of having just 3 months of registries, which is not usually advised, due to issues related with incomplete or short training. For the same reason, this dataset has not sufficient data to allow the study of clustering based forecasts, neither to study the approaches involving weekly periodicity or 70 past observations.

The results achieved by the best models in each approach are presented in Table 3. It can be observed the consistency of the KNN methodology, in particular when it is configured with 2 neighbors and the Euclidian distance weight function, since this model is found to give the best results in every approach. It is also observable that using a 12h time window when training and forecasting offers the best performance considering any of the metrics presented.

Table 3 - RMSE, MAPE% and R2 of the best model found with each approach tested for the Cars Benchmark.

Periodicity	Features	Model	RMSE (-)	MAPE (%)	R^2
24h	Demand (3)	KNN(N=2 euclidian)	0.1150	35.0550	0.5199
24h	Demand (14)	KNN(N=2 euclidian)	0.0922	36.4973	0.6185
12h	Demand (14)	KNN(N=2 euclidian)	0.0553	25.5196	0.8108
48h	Demand (14)	KNN(N=2 euclidian)	0.1475	59.9765	0.4100

In Figure 4 it is visible that the data presents a very atypical behavior. This represents a great difficulty when training the models, as each new sample can potentially bring more noise with no contribution to the process, and also for the prediction phase, as the new data has a high probability of being something the models have not previously been confronted with.

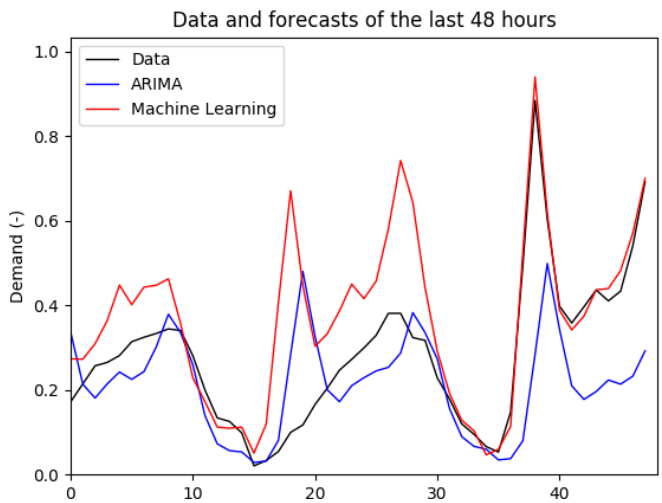

Figure 4 - Cars Benchmark . Water consumption observed in the last 2 days. Forecasts given by the ARIMA and KNN(N=2 euclidian) models.

The results obtained by the best model of each family are compared in Table 4, where the ARIMA is also included for comparison purposes. Except for the SVR, every other methodology presents at least one model that is better than the ARIMA considering any metric. As for the SVR, it gets particularly bad results in this benchmark, with its best model presenting RMSE and MAPE% about twice as bad as the best models in the other methodologies. Overall, the 2 best models are the KNN with Euclidian distance weight function. Note that the difference between the best and the second best models is much more accentuated that that between the second and the third best models.

Even though the SVR methodology did not achieve the expectations, one can conclude that using machine learning techniques proves to outperform the ARIMA in this benchmark.

Table 4 - Models that achieved the (i) best RMSE per family, (ii) the overall best 3 RMSE and (iii) the overall worst RMSE, and (iv) the ARIMA results, applied to the Cars Benchmark, using 14 12h demand samples.

Model	RMSE (-)	MAPE (%)	R^2
RFR(N=8 n=2)	0.0769	35.5161	0.6288
KNN(N=2 euclidian)	0.0553	25.5197	0.8108
SVR(linear t=0.001)	0.1264	68.2466	0.0076
ANN(relu(2x75) lbfgs)	0.0637	29.3742	0.7841
KNN(N=2 euclidian)	0.0398	25.5197	0.8108
KNN(N=5 euclidian)	0.0637	29.0592	0.7623
ANN(relu(2x75) lbfgs)	0.0637	29.3742	0.7841
ANN(identity(5x75) adam)	0.3128	147.5146	-0.4758
ARIMA	0.1255	48.2089	0.2121

4.2.3. Results for the the Air Quality Benchmark

For this benchmark the available data allowed more tests to be made, including tests using weather features, which have not been made previously. Therefore, adding to the tests presented in the first benchmark, 2 tests using temperature features were also made. 14 temperature features were considered for one test and just 1 temperature feature was considered for the other. Both consider a periodicity of 24h and 14 demand features.

Table 5 confirms that using periodicity of 12 hours brings the best results. It also shows that for this benchmark's database, the best methods are Neural Networks. The use of weather features did not bring an improvement in the performance and the increase of the periodicity clearly improves the R^2, but not the RMSE nor the MAPE%.

Table 5 - RMSE, MAPE% and R^2 of the best model found with each approach tested for the Air Quality Dataset.

Periodicity	Features	Model	RMSE (-)	MAPE (%)	R^2
24h	Demand (3)	ANN(identity(2x75) sgd)	0.0705	34.9673	-3.4301
24h	Demand (14)	ANN(identity(2x25) adam)	0.0723	35.4680	-0.0730
24h	Demand (70)	ANN(identity(8x10) lbfgs)	0.0664	44.9123	0.0571
12h	Demand (14)	ANN(identity(8x10) lbfgs)	0.0655	35.8837	-0.3904
48h	Demand (14)	ANN(identity(8x25) lbfgs)	0.0884	51.4168	0.1180
168h	Demand (14)	ANN(relu(2x10) adam)	0.0770	30.4780	0.5630
24h	Demand (14), Temperature (14)	SVR(linear t=0.01)	0.1157	37.7114	0.0338
24h	Demand (14), Temperature (1)	ANN(relu(2x10) adam)	0.1134	36.0354	-0.3135

The best models in each method are shown in Table 6. This table allows to conclude that for this benchmark the R^2 achieved are particularly low. This suggests that training the models with the objective of maximizing R^2 would probably bring better overall results, supported by the fact that some models produce forecasts with a much better R^2 with little prejudice to the

RMSE (3rd and 4th entries in Table 5). However, the RMSE results are approximately those observed previously. In this benchmark it is notable that the range between the best and the worst models' results is less than 40% of the worst RMSE. As before, machine learning methods proved to find better solutions than the ARIMA. Figure 5 illustrates the best machine learning model (ANN identity(8x10) lbfgs) in comparison with the ARIMA for the Air Quality Benchmark. The ARIMA model shows a tendency of over estimating the real demand.

Table 6 - Models that achieved the (i) best RMSE per family, (ii) best 3 RMSE overall and (iii) worst RMSE overall, and (iv) ARIMA results, applied to the Air Quality Benchmark, using 14 12h demand features.

Model	RMSE (-)	MAPE (%)	R^2
RFR(N=5 n=8)	0.0800	47.4656	-0.2735
KNN(N=8 euclidian)	0.0772	45.8370	-0.0914
SVR(rbf t=0.01)	0.0680	40.1903	-0.5240
ANN(identity(8x10) lbfgs)	0.0655	35.8837	-0.3904
ANN(identity(8x10) lbfgs)	0.0655	35.8837	-0.3904
ANN(identity(8x25) lbfgs)	0.0666	37.2693	-0.9076
ANN(identity(2x10) lbfgs)	0.0672	37.5903	-0.5068
ANN(logistic(8x75) adam)	0.1082	60.2676	-2.9964
ARIMA	0.0919	54.5420	-1.2770

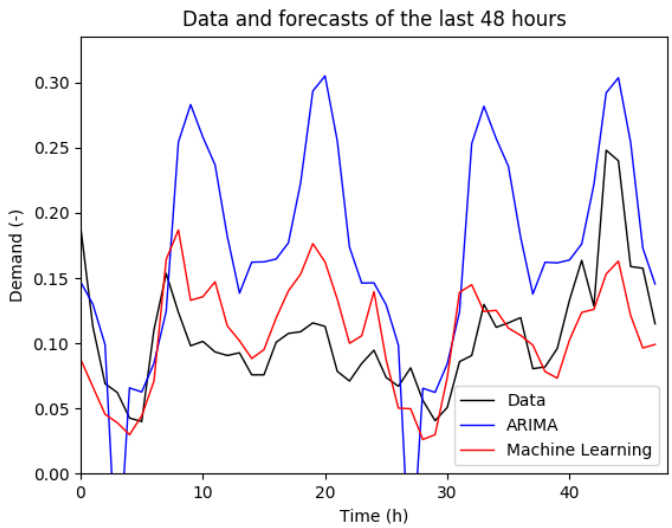

Figure 5 – Air Quality Benchmark. Water consumption observed in the last 2 days. Forecasts given by the ARIMA and ANN(identity(8x10) lbfgs) models.

The benchmark tests allowed the conclusion that the developed algorithm, specifically the machine learning strategy, is capable of producing predictions based on the previous observations and existing patterns in the data. In most cases, the machine learning methods can produce more accurate forecasts than ARIMA, which is a method often used in forecasting.

However, for different datasets, the best forecasts are often produced by different models. For this reason, it is always important to test different methods and models when a new database is being analyzed.

5. Applying the machine learning algorithms to water supply systems

The models previously described are applied to three databases provided by two Portuguese water utilities. Both companies store their data in similar ways. The cumulative amount of water that passes through any node of its network is saved, meaning the water demand in a determined period is the difference between the cumulative data observed at the extremities of that interval. The data was provided in raw, requiring a filtering and treatment step.

5.1. Sources of data

The first water utility – Water Utility 1 –is located in the north part of Portugal and is responsible for the water collection, treatment and distribution in an area of more than 2.500 km^2 serving more than 1.5 million people. This company provided data concerning 4 points of its network, but due to the errors found (in quantity and quality), only 2 are used in this work – WD2 and WD4. Visual representations of the WD2 and WD4 data (the last 48 hours) can be seen in Figure 6 and Figure 7, respectively.

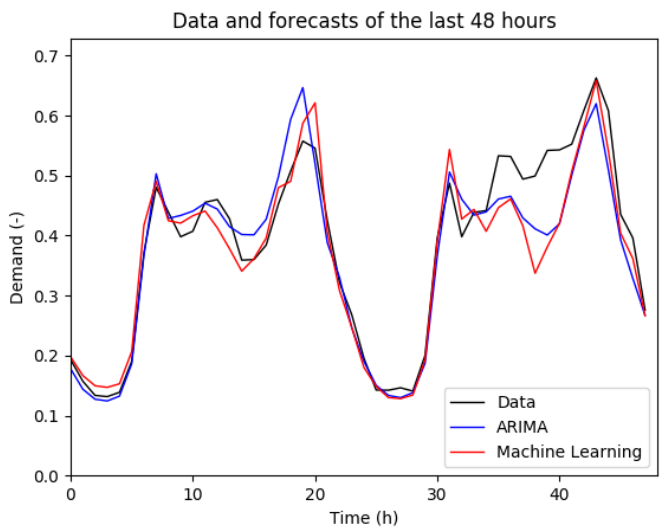

Figure 6 – WD2 database of Water Utility 1. Water consumption observed in the last 2 days. Forecasts given by the ARIMA and ANN(identity(8x10) lbfgs) models.

The second real data comes from a water utility located in central Portugal – Water Utility 2 –, responsible for supplying water to over 20 thousand customers. This company provided data referent to its entire network but with an evident lack of data in some points. In other points of the network, the existent data shows excessive errors. For this reason, only the data of one

point of the network will be considered to train the models. The last 48h of this dataset are represented in Figure 8.

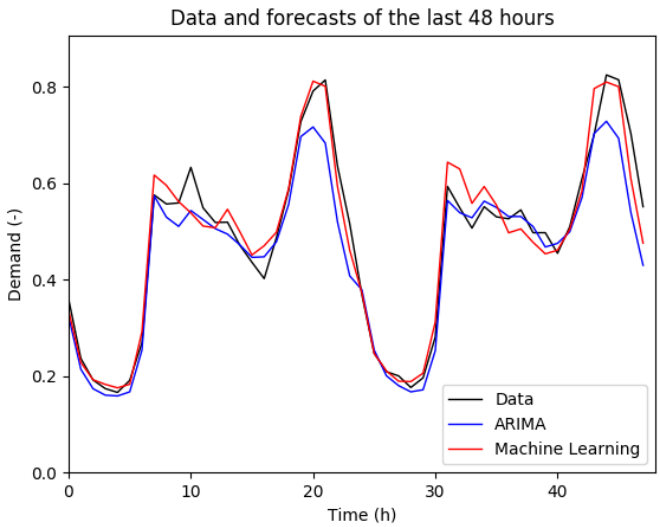

Figure 7 - WD4 database of Water Utility 1. Water consumption observed in the last 2 days. Forecasts given by the ARIMA and ANN(relu(8x25) lbfgs) models.

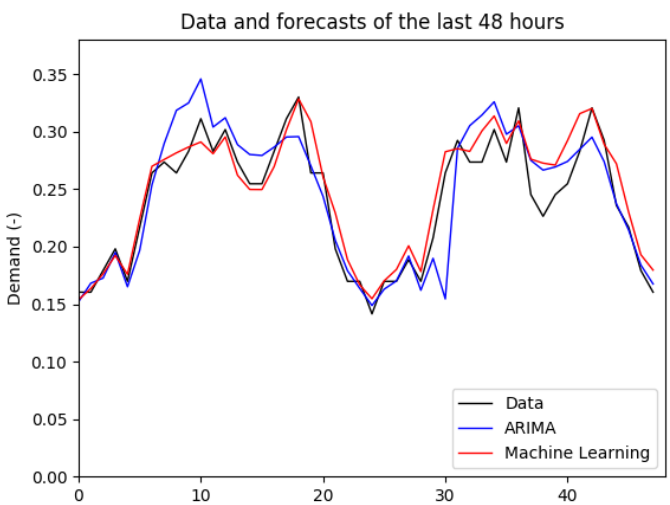

Figure 8 - Water Utility 2 database. Water consumption observed in the last 2 days. Forecasts given by the ARIMA and ANN(identity(2x10) lbfgs) models.

The observation of the data given by both companies allows to identify a few problems regarding either the presence of outliers or the absence of data. It was considered as outliers all

the values that did not fall in the range between the average and a margin of 3 times the standard deviation. The value of the upper or lower limit of the described range was assigned to the outliers, respectively if they lied above or under the boundaries. When no values were found at any given instant, the algorithm assigned the global average to those instances. A second iteration of this process (removing the outliers followed by assigning the average to missing values) is applied, to reduce the impact of the errors detected before the first iteration. After the correction of all outliers and missing values, a normalization sub-routine is applied. It divides each value in the data (demand, temperature or rain occurrence) by the corresponding maximum value. Therefore, each variable become dimensionless and consequentially has the same relative importance.

5.2. Results for the Water Utility 1

The 99 designed models were evaluated according to 11 approaches, including one considering clustering, two considering temperature history and two considering rain occurrence history.

The results obtained by the best model for each periodicity and features approach are presented in Table 7. The results obtained when applied to real water demand data are better than those obtained for other phenomena, although worse than those obtained for benchmark 1. This statement is supported by the MAPE% and R^2 improvement, even though the RMSE often suffered a small decrease. Since the forecasting algorithm was written with the water demand forecasting specifications in mind this fact is expected. The 12h periodicity gives the best RMSE (the objective function) by a small margin, but also presents the worst R^2 and the 3rd worst MAPE%. However, because the fitting process is performed using the RMSE, that must be the metric to consider when comparing the models' performance. Therefore, the best model is the ANN(identity(8x10) lbfgs) with a periodicity of 12h and 14 demand features. Increasing the periodicity or decreasing the number of features in the forecasts worsens their RMSE results. Concerning the weather features, using temperature or rain occurrence features presents the same results, suggesting a high correlation between the temperature and the occurrence of rain in any specific period. Using less weather features brings benefits to the forecasts. Consequentially, it is advisable not to use weather features. The best models are Neural Networks with the LBFGS learning algorithm with identity or rectified linear unit activation function.

In Table 8 the best results obtained by each method are presented, considering no weather features, and 14 registries of past water demand in 12h intervals. A deeper look reveals that the KNN models' performance has a clear tendency of improving with the number of neighbors considering all metrics. The results show no significant difference between the weight functions tested, although the results are slightly better when using the Euclidian distance. The SVR method is less sensible to the tolerance used, since changing that parameter has an insignificant impact in any metric, across all approaches. The selection of the kernel appears to be specific to each approach, since no particular kernel is consistently the best solution. At the same time, no particular kernel presents particularly bad results. Nonetheless, the kernel has a bigger importance in the forecasts than the tolerance. Concerning the RFR models, expanding the size of the forest (number of trees) has a positive impact in the quality of the forecasts. The same can be said about the number of required samples at each split. RFR(N=8 n=8) is the best RFR in most approaches. The worst 12 ANN models use SGD, and of those, 9 use the logistic activation function. The 12 best ANN models use the LBFGS learning algorithm and none of

them uses the logistic activation function. Therefore, it is advised not to use the SGD and logistic in comparison to the LBFGS. The shape of the network has a smaller importance in the outcome of the forecasts, but smaller networks seem to result in better results. All methods presented better forecasts than the ARIMA (considering RMSE). Figure 6 presents the forecasts made by the ARIMA and ANN(identity(8x10) lbfgs) models in this dataset.

Table 7 – RMSE, MAPE% and R2 of the best model found with each approach using the WD2 database of the Water Utility 1.

Periodicity	Features	Model	RMSE (-)	MAPE (%)	R^2
24h	Demand (3)	ANN(identity(8x10) lbfgs)	0.0669	11.1843	0.7582
24h	Demand (14)	ANN(relu(2x75) lbfgs)	0.0554	10.0918	0.8534
24h	Demand (70)	ANN(identity(8x25) lbfgs)	0.0561	9.7884	0.8609
12h	Demand (14)	ANN(identity(8x10) lbfgs)	0.0532	10.8467	0.7331
48h	Demand (14)	ANN(identity(8x10) lbfgs)	0.0605	10.7018	0.8322
168h	Demand (14)	ANN(relu(2x25) adam)	0.0590	12.0670	0.8700
24h	Demand, using clustering (14)	ANN(identity(5x10) lbfgs)	0.0624	11.2067	0.8040
24h	Demand (14), Temperature (14)	ANN(relu(5x10) lbfgs)	0.0684	10.2677	0.8564
24h	Demand (14), Temperature (1)	ANN(relu(5x10) lbfgs)	0.0678	9.5399	0.8567
24h	Demand (14), Rain Occurrence (14)	ANN(relu(5x10) lbfgs)	0.0684	10.2677	0.8564
24h	Demand (14), Rain Occurrence (1)	ANN(relu(5x10) lbfgs)	0.0678	9.5399	0.8567

Table 8 – Models that achieved the (i) best RMSE per family, (ii) best 3 RMSE overall and (iii) worst RMSE overall, and (iv) ARIMA results, applied to the WD2 database of Water Utility 1, using 14 12h demand features.

Model	RMSE (-)	MAPE (%)	R^2
RFR(N=8 n=8)	0.0594	11.1305	0.6854
KNN(N=8 uniform)	0.0556	10.8876	0.7150
SVR(linear t=0.01)	0.0543	11.4832	0.6552
ANN(identity(8x10) lbfgs)	0.0532	10.8467	0.7331
ANN(identity(8x10) lbfgs)	0.0532	10.8467	0.7331
ANN(identity(2x25) lbfgs)	0.0533	10.9579	0.7194
ANN(identity(8x25) lbfgs)	0.0536	10.6897	0.7220
ANN(logistic(5x10) sgd)	0.1677	48.3851	-832.8060
ARIMA	0.0644	10.8487	0.8390

When forecasting the water demand in the second network, the best results are found for a periodicity of 24h and 14 demand features. In this case, the use of 14 weather features seems to be better than the case of using just one, but worse than the case that does not use this feature. The Neural Network models continue to present the best performances.

Analyzing the individual models one can confirm the tendency previously observed. The best models found with this dataset show slightly better results than those found using WD2. By comparing Table 10 with Table 8 one can also observe that the best models seem independent of the dataset used. Namely, using the Euclidian weight function in KNN models

with 8 neighbors, the rectifier or identity activation functions combined with LBFGS learning algorithm in ANN models and 8 estimators with 8 samples in each split in RFR models consistently presents good forecasts. For this case, the ARIMA presents slightly better RMSE than machine learning methods. However, the best ANN presents a much better MAPE% and R^2 with little prejudice of the RMSE. Figure 7 shows the forecasts made by ANN(relu(8x25) lbfgs) and the ARIMA models.

5.3. Results for the Water Utility 2

A similar analysis can be made for the second dataset. Generically, the RMSE and MAPE% found with this dataset are better than those found for the Water Utility 1 datasets, while the R^2 drops, as seen in Table 11 in comparison to Table 7 and Table 9. The use of 14 samples of 24h presents the best RMSE and MAPE% results. Using clusters in the forecasts does not bring better forecasts, whichever the dataset, but occasionally results in a better correlation between the forecasts and the observations. The use of similar days to train the models results in a more correctly identified pattern, but also results in fewer examples available for training, possibly resulting in fewer iterations and incomplete training.

Table 9 - RMSE, MAPE% and R2 of the best model found with each approach using the WD4 database of the Water Utility 1.

Periodicity	Features	Model	RMSE (-)	MAPE (%)	R^2
24h	Demand (3)	ANN(identity(5x25) sgd)	0.0762	12.7028	0.8226
24h	Demand (14)	ANN(relu(8x25) lbfgs)	0.0473	7.8511	0.9229
24h	Demand (70)	ANN(identity(2x25) lbfgs)	0.0546	8.9986	0.9080
12h	Demand (14)	ANN(identity(8x10) lbfgs)	0.0490	8.5004	0.8143
48h	Demand (14)	ANN(relu(5x25) lbfgs)	0.0575	9.5678	0.8995
168h	Demand (14)	ANN(identity(2x25) lbfgs)	0.0610	10.2320	0.8980
24h	Demand, using clustering (14)	ANN(identity(8x10) sgd)	0.0590	10.1124	0.8937
24h	Demand (14), Temperature (14)	ANN(identity(8x10) lbfgs)	0.0523	7.8589	0.9271
24h	Demand (14), Temperature (1)	ANN(relu(2x10) lbfgs)	0.0530	8.0717	0.9240
24h	Demand (14), Rain Occurrence (14)	ANN(identity(8x10) lbfgs)	0.0523	7.8589	0.9271
24h	Demand (14), Rain Occurrence (1)	ANN(relu(2x10) lbfgs)	0.0530	8.0717	0.9240

The best models of each family of methods are presented in Table 12. Surprisingly, the SVR methods did not present identical results to those obtained previously. However, note that the results obtained by the different methods have a smaller range than those observed using the previous datasets. The best model and the ARIMA's forecasts are represented in Figure 8.

Table 10 - Models that achieved the (i) best RMSE per family, (ii) best 3 RMSE overall and (iii) worst RMSE overall, and (iv) ARIMA results, applied to the WD4 database of Water Utility 1, using 14 24h demand features.

Model	RMSE (-)	MAPE (%)	R^2
RFR(N=5 n=2)	0.0623	9.6819	0.8799
KNN(N=8 euclidian)	0.0681	10.4101	0.8548
SVR(rbf t=0.001)	0.0574	9.2291	0.8814
ANN(relu(8x25) lbfgs)	0.0473	7.8511	0.9229
ANN(relu(8x25) lbfgs)	0.0473	7.8511	0.9229
ANN(relu(5x10) lbfgs)	0.0474	8.0949	0.9293
ANN(relu(5x25) lbfgs)	0.0483	8.4529	0.9101
ANN(relu(5x10) sgd)	0.2359	40.1780	-79.0492
ARIMA	0.0417	8.5338	0.8659

Table 11 – RMSE, MAPE% and R^2 of the best model found with each approach using the Water Utility 1 database.

Periodicity	Features	Model	RMSE (-)	MAPE (%)	R^2
24h	Demand (3)	KNN(N=8 uniform)	0.0315	9.3134	0.7067
24h	Demand (14)	ANN(identity(2x10) lbfgs)	0.0228	7.0477	0.7532
24h	Demand (70)	ANN(identity(5x75) lbfgs)	0.0268	8.6002	0.6626
12h	Demand (14)	ANN(identity(2x25) lbfgs)	0.0242	7.8026	0.6314
48h	Demand (14)	ANN(relu(2x10) lbfgs)	0.0276	8.2274	0.7314
168h	Demand (14)	ANN(relu(2x10) lbfgs)	0.0340	10.117	0.6920
24h	Demand, using clustering (14)	ANN(identity(8x10) lbfgs)	0.0266	7.9235	0.7420

Table 12 – Models that achieved the (i) best RMSE per family, (ii) best 3 RMSE overall and (iii) worst RMSE overall, and (iv) ARIMA results, applied to the Water Utility 2 database, using 14 24h demand features.

Model	RMSE (-)	MAPE (%)	R^2
RFR(N=8 n=8)	0.0264	8.1820	0.6849
KNN(N=8 euclidian)	0.0273	8.2914	0.6450
SVR(linear t=0.001)	0.0578	23.9645	-0.5905
ANN(identity(2x10) lbfgs)	0.0228	7.0477	0.7532
ANN(identity(2x10) lbfgs)	0.0228	7.0477	0.7532
ANN(identity(5x25) lbfgs)	0.0239	7.3169	0.7338
ANN(identity(5x10) lbfgs)	0.0242	7.4216	0.7385
ANN(logistic(5x75) adam)	0.0814	32.2616	-1.1143
ARIMA	0.0452	8.6048	0.5784

5.4. Results for the Parallel Strategies

Taking into account the results afore discussed and the ones found in the literature, one can assess which model configurations and forecasting techniques might present the best results. Alternatively of designing a model presumed to accomplish good results across different databases, it is possible to conceive a pool of models and approaches which combination outperforms each individual model. The analysis made so far show that the best models should respect the following criteria:

- 24h forecast window;
- Approximately 2 weeks of previous water demand observations;
- When configuring ANN:
  - LBFGS learning algorithm;
  - Rectifier or identity activation function;
  - Small networks;
- When configuring KNN:
  - Euclidian weight function;
  - 8 neighbors;
- When configuring RFR:
  - 8 trees per forest;
  - 8 or more samples at each split.

The new pool of models being tested is composed by 4 RFR, 3 KNN, 3SVR and 12 ANN. The RFR models have 5 or 8 trees per forest and 2 or 8 minimum samples per split. The KNN models use the Euclidian distance weight function for 7, 8 or 9 neighbors (refining the previous numbers of neighbors tested). The SVR uses the 3 kernels tested so far, with the tolerance of 0.01. The ANN uses the LBFGS learning algorithm with identity or rectifier activation functions, with the 10 or 25 neurons distributed by 2, 5 or 8 layers.

The hybrid method with parallel forecasts is applied to the databases concerning the Water Utility 1 and Water Utility 2. For this case, no temperature or rain occurrence data is necessary. However, when applied to the Water Utility 2 database, the models tested only present Mean Errors above zero. The hybrid method performance cannot be tested in that dataset.

Table 13 compiles the results obtained by the models which output is used as a parcel of the weighted average forecast in the two datasets and their corresponding weights. The proposed hybrid methodology results in a significant improvement in RMSE with a slight decrease in MAPE% and R^2 of the best models in WD2. It also improved the MAPE% with zero prejudice of the RMSE in WD4, although it cost a small decrease in R^2. The Mean Error of the forecasts using the hybrid strategy is zero. Generically, it is safe to use the presented hybrid methodology, assuming the models used in the parallel computations satisfy a set of pre-requisites relative to their expected performance. Figure 9 represents the observed demand of the last 48h of WD2 in water Utility 1, as well as the forecasts made by HPM, ANN(relu(2x10) lbfgs) and RFR(N=5 n=2). Figure 10 represents the demand of the last 48h of WD4 in Water Utility 1, and the respective forecasts using the HPM, ANN(relu(5x25) lbfgs) and ANN(relu(2x25) lbfgs) models.

Table 13 – Results obtained using the hybrid parallel methodology (HPM) when applied to WD2 and WD4 of Water Utility 1. The results obtained by the models used for the weighted average are also presented.

Dataset	Model	Weight (%)	ME (-)	RMSE (-)	MAPE (%)	R^2
WD2	ANN(relu(2x10) lbfgs)	24.22	-0.0062	0.0689	9.8159	0.8600
	ANN(relu(2x10) lbfgs)	24.22	-0.0062	0.0689	9.8159	0.8600
	ANN(relu(2x10) lbfgs)	24.22	-0.0062	0.0689	9.8159	0.8600
	SVR(rbf t=0.01)	7.81	0.0020	0.7190	11.8694	0.8204
	SVR(rbf t=0.01)	7.81	0.0020	0.7190	11.8694	0.8204
	RFR(N=5 n=2)	11.72	0.0030	0.7820	11.2684	0.8126
	HPM	100	0.0000	0.0490	10.2249	0.8438
WD4	ANN(relu(5x 25) lbfgs)	13.33	-0.0014	0.0544	8.0566	0.9261
	ANN(relu(5x 25) lbfgs)	13.33	-0.0014	0.0544	8.0566	0.9261
	ANN(relu(5x25) lbfgs)	13.33	-0.0014	0.0544	8.0566	0.9261
	ANN(relu(8x25) lbfgs)	16.19	0.0017	0.0537	8.4290	0.9279
	ANN(relu(8x25) lbfgs)	16.19	0.0017	0.0537	8.4290	0.9279
	ANN(relu(2x25) lbfgs)	27.63	0.0029	0.0577	8.3888	0.9101
	HPM	100	0.0000	0.0537	7.9281	0.9267

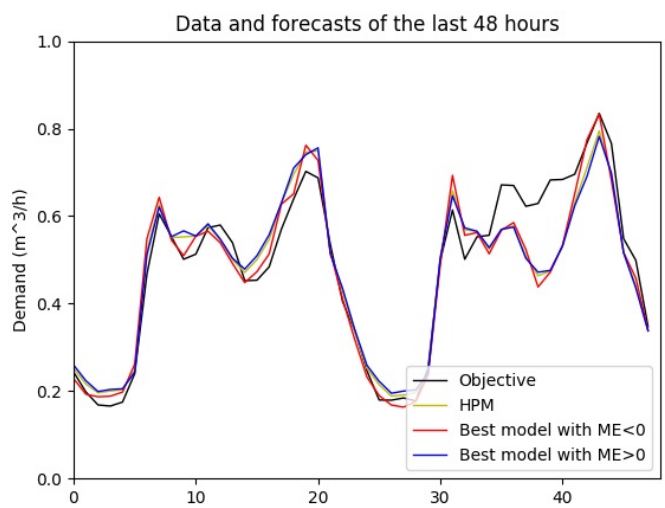

Figure 9 - WD2 database of Water Utility 1. Water consumption observed in the last 2 days. Forecasts given by the HPM, ANN(relu(2x10) lbfgs) and RFR(N=5 n=2) models.

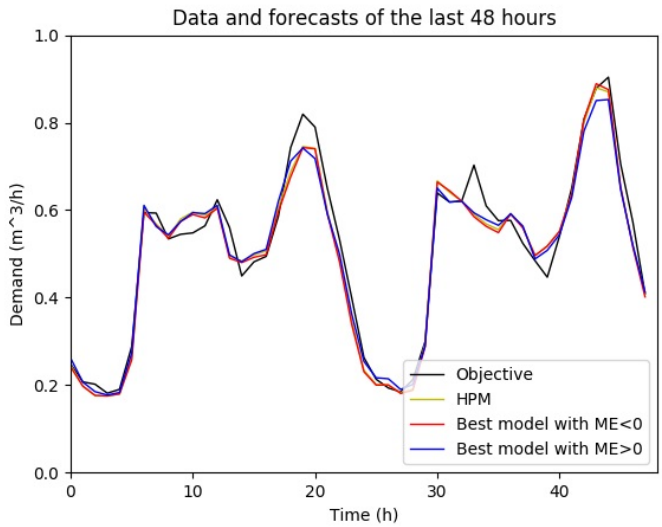

Figure 10 - WD4 database of Water Utility 1. Water consumption observed in the last 2 days. Forecasts given by the HPM, ANN(relu(5x25) lbfgs) and ANN(relu(2x25) lbfgs) models.

6. Conclusions

This work presents a Machine Learning water demand forecasting strategy that is capable of producing accurate predictions when compared to traditional strategies. It was found to be reliable when applied to real data, provided no significant anomalies of the data used during training. The statistical metrics here discussed support the fact that the forecasts made are similar to the real observation independently of the time of the day in cause. For these reasons it can concluded that the developed algorithm can be applied to real cases as the forecast module for decision support systems in water utilities equipment management. The forecasts presented by this module do not provide the optimal operation schedule of the equipment. For that, further works must be done. Studies presented in section 2.6 indicate that developing and applying such algorithm may result in cost reduction of 18% or more [10], [11] and [12].

Nonetheless, some remarks on the use of the presented algorithm arise. Although it was found that the best results are consistently given by the same group of models it is not guaranteed that for new data those models will maintain its performance. When applying the algorithm in different datasets, a large set of models must be trained in order to infer the most appropriate models. If applied to real cases where new data is constantly being acquired it is important that the models are retrained on a regular basis (as frequently as possible, provided the computational power for that is available). Note that in the latter case, the introduction of new data could mean that the accuracy of the models that were previously found to be the most adequate for that specific network is affected. Consequentially, the periodic retraining suggested must include the larger set of models. Additionally, the proposed hybrid parallel methodology proved its usefulness (around 15% improvement in RMSE) and should be used when possible.

According to the tests made, machine learning methods should be chosen over traditional time series analysis. Although the ARIMA often provides results better than those achieved by some machine learning models most of the time there are other machine learning models that outperform ARIMA (about 18% in RMSE and 8% in MAPE%). That said, both strategies should be tested in order to assess their real value in the case being studied.

References

[1] Águas de Trás-os-Montes e alto Douro, "Perguntas Frequentes," [Online]. Available: http://www.aguas-tmad.pt/pt/versao-didactica/perguntas-frequentes/categoria/sobre-a-adtmad/#11. [Accessed 12 07 2017].

[2] B. d. C. Coelho, "Energy efficiency of water supply systems using optimization techniques and micro-hydroturbines," Aveiro, Portugal, 2016.

[3] S. M. Bunn and L. Reynolds, "The energy-efficient benefits of pump-scheduling optimization for potable water supplies," *IBM Journal of Research and Develpment*, vol. 53, no. 3, pp. 5:1 - 5:13, 2009.

[4] T. M. Mitchell, Machine Learning, Boston: McGraw-Hill, 1997.

[5] C. M. Bishop, Pattern Recognition and Machine Learing, New York: Springer, 2006.

[6] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference for Learning Representations*, San Diego, 2015.

[7] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503-528, 1989.

[8] The Pennsylvania State University, "The Coefficient of Determination, r-squared," PennState Eberly College of Science, 2017. [Online]. Available: https://onlinecourses.science.psu.edu/stat501/node/255. [Accessed 07 07 2017].

[9] R. J. Hyndman, "Hyndsight blog," 25 8 2010. [Online]. Available: https://robjhyndman.com/hyndsight/benchmarks/. [Accessed 15 6 2017].

[10] H.-S. Kang, H. Kim, J. Lee, I. Lee, B.-Y. Kwak and H. Im, "Optimization of Pumping Schedule Based on Water Demand Forecasting Using Combined Model of Autorregressive Integrated Moving Average and Exponential Smoothing," *Water Science & Technology Water Supply*, vol. 15, no. 1, pp. 188-195, 2014.

[11] E. Salomons, A. Goryashko, U. Shamir, Z. Rao and S. Alvisi, "Optimizing the operation of the Haifa-A water-distribution network," *Journal of Hydroinformatics*, vol. 09, no. 1, pp. 51-64, 2007.

[12] G. Cembrano, G. Wells, J. Quevedo, R. Pérez and R. Argelaguet, "Optimal Control of a Water Distribution Network in a Supervisory Control System," *Control Engineering Practice*, vol. 8, no. 10, pp. 1177-1188, 2000.

[13] A. Cendelieri, D. Conti, D. Cappellini and F. Archetti, "Urban Water Demand Characterization And Short-Term Forecasting - The ICeWater Project Approach," in *International Conference on Hydroinformatics*, New York, 2014.

[14] S. Alvisi, M. Franchini and A. Marinelli, "A short-term, pattern-based model for water-demand forecasting," *Journal of Hydroinformatics*, vol. 09, no. 1, pp. 39-50, 2007.

[15] M. Bakker, J. H. Vreeburg, K. M. van Schagen and L. C. Rietveld, "A fully adaptive forecasting model for short-term drinking water demand," *Environmental Modelling & Software*, vol. 48, pp. 141-151, June 2013.

[16] A. Candelieri, D. Soldi, D. Conti and F. Archetti, "Analytical Leakages Localization in Water Distribution Networks Through Clustering and Support Vector MACHINES. The Icewater Approach," *Procedia Engineering*, vol. 89, pp. 1080-1088, 2014.

[17] M. Herrera, L. Torgo, J. Izquierdo and R. Pérez-Garcia, "Predictive models for forecasting hourly urban water demand," *Journal of Hydrology*, vol. 387, pp. 141-150, April 2010.

[18] J. D. de Lima, G. O. Adamczuk, M. G. Trentin, D. R. Batistus and C. B. Pozza, "A Study of the Performance of Individual Techniques and Their Combinations to Forecast Urban Water Demand," *Revista Espacios*, vol. 37, no. 22, pp. 5-28, 14 04 2016.

[19] M. Tiwari, J. Adamowski and K. Adamowski, “Water Demand Forecasting Using Extreme Learning Machines,” *Journal of Water and Lanc Development*, vol. 25, no. 1-3, pp. 37-52, 2016.

[20] C. Peña-Guzman, J. Melgarejo and D. Prats, “Forecasting Water Demand in Residential, Commercial, and Industrial Zones in Bogotá, Colombia, Using Least-Squares Support Vector Machines,” *Mathematical Problems in Engineering*, vol. 2016, 05 10 2016.

[21] M. Ghiassi, F. Fa'al and A. Abrishamchi, “Large metropolitan water demand forecasting using DAN2, FTDNN, and KNN models: A case study of the city of Tehran, Iran,” *Urban Water Journal*, 06 09 2016.

[22] B. M. Brentan, E. Luvizotto Jr., M. Herrera, J. Izquierdo and R. Pérez-García, “Hybrid Regression Model for Near Real-time Urban Water Demand Forecasting,” *Journal of computational and Applied Mathematics*, vol. 309, pp. 532-541, 02 02 2017.

[23] S. Shabani, P. Yousefi, J. Adamowski and G. Naser, “Intelligent Soft Computing Models in Water Demand Forecasting,” in *Water Stress in Plants*, Croatia, InTech, 2016, pp. 99-117.

[24] S. Moutadid and J. Adamowski, “Using Extreme Learning Machines for Short-term Urban Water Demand Forecasting,” *Urban Water Journal*, vol. 14, no. 6, pp. 360-368, 2017.

[25] M. M. Haque, A. de Souza and A. Rahman, “Water Demand Modelling Using Independent Component Regression Technique,” *Water Resources Management*, vol. 31, no. 1, pp. 299-312, 2017.

[26] V. Rodríguez-Galiano and M. C. Villarín-Clavería, “Regression Trees for Modelling Water Demand in Sevilla City, Spain,” in *Geostatistics and Machine Learning. Applications in Climate and Environmental Sciences*, Belgrade, 2016.

[27] N. Mellios, D. Kofinas, E. Papageorgiou and C. Laspidou, “A Multivariative Analysis of the Daily Water Demand of Skiathos Island, Greece, Implementing the Artificial Neuro-Fuzzy Inferente System (ANFIS),” in *E-proceedings of the 36th IAHR World Congress*, The Hague, Netherlands, 2015.

[28] D. Suh and S. Ham, “A Water Demand Forecasting Model using BPNN for Residential Building,” *Contemporary Engineering Sciences*, vol. 9, no. 1, pp. 1-10, 2016.

[29] Y. Seo, S. Kim, O. Kisi and V. P. Singh, “Daily water level forecasting using wavelet decomposition and artificial intelligence techniques,” *Journal of Hydrology*, vol. 250, pp. 224-243, 2015.

[30] J. Adamowski and C. Karapataki, “Comparison of Multivariative Regression and Artificial Neural Networks for Peak Urban Water-Demand Forecasting: Evaluation of Different ANN Learning Algorithms,” *Journal of Hydrologic Engineering*, vol. 15, pp. 729-743, October 2010.

[31] F. V. Veen, “The Neural Network Zoo,” The Asimov Institute, 16 09 2016. [Online]. Available: http://www.asimovinstitute.org/neural-network-zoo/. [Accessed 22 06 2017].

[32] D. Svozil, V. Kvasnicka and J. Pospíchal, “Introduction to Multi-layer Feed-forward Neural Networks,” *Chemometrics and Intelligent Laboratory systems*, vol. 39, pp. 43-62, 1997.

[33] R. H. Byrd, P. Lu, J. Nocedal and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal of Scientific Computing*, vol. 16, pp. 1190-1208, 1995.

[34] The Scipy community, “SciPy Reference Guide,” 21 06 2017. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/index.html. [Accessed 2017 07 08].

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[36] Python Software Foundation (US), “The Python Language Reference,” 14 6 2017. [Online]. Available: https://docs.python.org/3/reference/index.html. [Accessed 14 6 2017].

[37] Q. Qian and J. Pan, “A Creative Visualization of OLAP Cuboids,” 09 05 2017. [Online]. Available: http://www.ebaytechblog.com/2017/05/09/a-creative-visualization-of-olap-cuboids/. [Accessed 03 07 2017].

Appendix A. Benchmark Extensive Results

Table 14 - Best results per method and worst absolute results using Mathematically Generated Data Benchmark considering MAPE% and R^2. Periodicity=24h and 3 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	4.8933	0.8957
RFR(N=8 n=8)	4.9048	0.9012
KNN(N=8 uniform)	4.9105	0.9028
KNN(N=8 euclidian)	4.8996	0.9021
SVR(linear t=0.001)	5.2952	0.8880
ANN(identity(2x25) lbfgs)	4.7678	0.9086
ANN(logistic(2x10) sgd)	17.8162	-25.9280
ANN(logistic(5x10) sgd)	18.1475	-34.7719
ANN(logistic(8x10) sgd)	17.9813	-22.8276
ANN(logistic(2x25) sgd)	17.0605	-5.9263
ANN(logistic(5x25) sgd)	16.8960	-5.9366
ANN(logistic(8x25) sgd)	17.0499	-6.7851

Table 15 - Best results per method and worst absolute results using Mathematically Generated Data Benchmark considering MAPE% and R^2. Periodicity=24h and 70 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=2)	3.7452	0.9400
KNN(N=8 uniform)	3.4647	0.9475
KNN(N=8 euclidian)	3.4653	0.9476
SVR(linear t=0.001)	4.3049	0.9210
ANN(relu(5x25) lbfgs)	3.7358	0.9392
ANN(relu(5x75) lbfgs)	3.7246	0.9362
ANN(logistic(2x10) sgd)	17.4730	-44.0839
ANN(logistic(5x10) sgd)	17.8044	-59.4768
ANN(relu(8x10) sgd)	15.9487	-23.4598
ANN(logistic(8x10) sgd)	17.7742	-53.2311
ANN(logistic(2x25) sgd)	16.8523	-11.2013
ANN(logistic(5x25) sgd)	17.0026	-11.9283
ANN(relu(8x25) sgd)	16.0042	-15.5304

Table 16 - Best results per method and worst five absolute results using Mathematically Generated Data Benchmark considering MAPE% and R^2. Periodicity=12h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=2)	4.3829	0.6403
RFR(N=5 n=8)	4.1953	0.6196
KNN(N=8 uniform)	3.9457	0.6687
KNN(N=8 euclidian)	3.9408	0.6677
SVR(linear t=0.001)	4.3020	0.5949
ANN(relu(2x25) adam)	4.1008	0.6741
ANN(logistic(2x10) sgd)	17.8106	-151.0359
ANN(relu(5x10) sgd)	17.9570	-86.4887
ANN(logistic(5x10) sgd)	17.808	-158.1697
ANN(logistic(8x10) sgd)	17.8274	-188.7468
ANN(logistic(2x25) sgd)	17.6624	-39.2920
ANN(relu(5x25) sgd)	17.3699	-140.5075

Table 17 - Best results per method and worst five absolute results using Mathematically Generated Data Benchmark considering MAPE% and R^2. Periodicity=48h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	4.1214	0.9331
KNN(N=8 uniform)	3.8536	0.9411
SVR(linear t=0.001)	4.6481	0.9111
ANN(relu(5x10) lbfgs)	4.0133	0.9344
ANN(relu(5x75) lbfgs)	4.0872	0.9345
ANN(logistic(2x10) sgd)	18.3277	99.3478
ANN(relu(5x10) sgd)	18.8983	-43.6601
ANN(logistic(5x10) sgd)	18.7254	-117.7728
ANN(logistic(8x10) sgd)	18.7479	-137.0366
ANN(logistic(5x25) sgd)	18.3492	-29.5479
ANN(logistic(8x25) sgd)	18.3873	-31.2551

Table 18 - Best results per method and worst five absolute results using Mathematically Generated Data Benchmark considering MAPE% and R^2. Periodicity=168h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	3.7033	0.9493
KNN(N=8 euclidian)	3.6170	0.9510
SVR(linear t=0.01)	4.0977	0.9387
ANN(relu(5x10) lbfgs)	3.6940	0.9497
ANN(identity(8x10) lbfgs)	3.6430	0.9493
ANN(logistic(2x10) sgd)	17.8690	-55.0183
ANN(relu(5x10) sgd)	20.9247	-32.4700
ANN(logistic(5x10) sgd)	18.2723	-63.1830
ANN(logistic(8x10) sgd)	18.3000	-69.3400
ANN(logistic(8x25) sgd)	18.0740	-21.9490
ANN(identity(5x75) adam)	21.1633	0.5157
ANN(identity(2x75) adam)	27.9790	0.0830

Table 19 - Best results per method and worst five absolute results using Mathematically Generated Data Benchmark considering MAPE% and R^2. Periodicity=24h and 14 demand samples, with clustering.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	8.4858	0.6997
KNN(N=8 uniform)	9.1716	0.6702
KNN(N=8 euclidian)	9.1774	0.6706
SVR(poly t=0.01)	7.0297	0.7841
SVR(poly t=0.001)	7.0385	0.7851
ANN(identity(2x10) sgd)	4.8732	0.8919
ANN(logistic(2x10) sgd)	17.7483	-26.8800
ANN(relu(25x10) sgd)	13.9670	-8.8907
ANN(logistic(5x10) sgd)	18.1659	-29.2406
ANN(logistic(8x10) sgd)	18.2040	-34.8070
ANN(logistic(8x25) sgd)	17.1563	-7.1338
ANN(logistic(5x25) sgd)	17.0453	-6.7228

Table 20 - Best results per method and worst five absolute results using the Cars Benchmark considering MAPE% and R^2. Periodicity=24h and 3 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=2)	40.5274	0.3337
RFR(N=8 n=2)	40.7583	0.4661
KNN(N=2 euclidian)	35.0550	0.5199
SVR(rbf t=0.001)	85.1459	-2.4501
ANN(relu(2x25) lbfgs)	55.2987	0.1434
ANN(relu(2x75) lbfgs)	54.5667	-0.0143
ANN(identity(2x10) sgd)	150.1130	-1.6006
ANN(identity(5x10) sgd)	137.9950	-5.2107
ANN(logistic(5x10) sgd)	121.2810	-13801.7090
ANN(logistic(8x10) sgd)	121.1630	-11172.1170
ANN(logistic(2x25) sgd)	129.4460	-27602.8290
ANN(logistic(5x25) sgd)	124.9140	-17442.4290
ANN(identity(8x25) sgd)	129.6450	-49333.314
ANN(logistic(2x75) adam)	136.8480	-21.0634
ANN(identity(5x75) adam)	186.7530	-0.8456
ANN(logistic(8x75) adam)	140.6060	-16.8447

Table 21 - Best results per method and worst five absolute results using the Cars Benchmark considering MAPE% and R^2. Periodicity=12h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=2)	36.0818	0.5807
RFR(N=8 n=2)	37.2854	0.6284
KNN(N=2 euclidian)	25.5197	0.8108
SVR(linear t=0.01)	64.7271	0.1424
SVR(linear t=0.001)	65.4703	0.1511
ANN(relu(5x25) lbfgs)	30.3709	0.7896
ANN(relu(2x75) lbfgs)	29.3742	0.7841
ANN(relu(2x10) sgd)	147.3490	-5.7761
ANN(logistic(2x10) sgd)	132.2230	-23246.9830
ANN(relu(5x10) sgd)	140.5410	-745.4334
ANN(logistic(5x10) sgd)	134.7830	-160132.6400
ANN(logistic(8x10) sgd)	136.9600	-108062.3500
ANN(logistic(5x25) sgd)	130.0480	-125043.8300
ANN(identity(8x25) sgd)	140.2480	-9.1311
ANN(logistic(8x25) sgd)	129.6300	-105916.3500
ANN(identity(5x75) adam)	147.5150	-0.4758

Table 22 - Best results per method and worst five absolute results using the Cars Benchmark considering MAPE% and R^2. Periodicity=48h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=2)	97.4760	-0.1495
RFR(N=8 n=8)	90.6525	-0.1655
KNN(N=2 euclidian)	59.9765	0.4100
SVR(linear t=0.001)	128.9710	-0.5590
SVR(rbf t=0.01)	116.2790	-2.1015
ANN(relu(5x75) adam)	85.8310	-0.2015
ANN(relu(5x75) lbfgs)	112.4750	0.1625
ANN(logistic(2x10) sgd)	157.9900	-5485.6985
ANN(identity(5x10) sgd)	186.1500	-1.4155
ANN(logistic(5x10) sgd)	161.5380	-34726.1570
ANN(identity(8x10) sgd)	179.0760	-0.3775
ANN(logistic(8x10) sgd)	175.2020	-3252.9090
ANN(logistic(5x25) sgd)	173.5920	-1370.9685
ANN(identity(8x25) sgd)	202.7230	-3.536
ANN(logistic(8x25) sgd)	158.7740	-4218.8915
ANN(identity(5x75) adam)	210.7440	-0.5835

Table 23 - Best results per method and worst five absolute results using the Air Quality Benchmark considering MAPE% and R^2. Periodicity=24h and 70 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	60.8625	-0.1111
RFR(N=8 n=8)	51.5933	-0.2566
KNN(N=2 euclidian)	61.6296	-0.2220
KNN(N=8 uniform)	49.4464	0.3608
SVR(linear t=0.01)	38.7717	-0.2816
SVR(rbf t=0.001)	40.7838	-0.1565
ANN(identity(2x75) sgd)	34.9673	-3.4301
ANN(identity(8x75) sgd)	58.9810	-0.004
KNN(N=2 uniform)	62.2627	-0.2275
ANN(identity(2x10) sgd)	59.4601	-2.2598
ANN(logistic(2x10) sgd)	45.7291	-320.6503
ANN(relu(5x10) sgd)	46.8687	-233.8830
ANN(logistic(5x10) sgd)	46.1277	-807.6013
ANN(identity(8x10) adam)	61.7747	-0.1257
ANN(logistic(8x10) sgd)	45.6381	-255.9002
ANN(logistic(5x75) sgd)	46.3458	-168.8472

Table 24 - Best results per method and worst five absolute results using the Air Quality Benchmark considering MAPE% and R^2. Periodicity=12h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	70.5971	0.0266
RFR(N=8 n=8)	54.9593	-0.0183
KNN(N=2 euclidian)	79.3250	-0.0853
KNN(N=8 uniform)	74.1552	-0.0957
SVR(linear t=0.01)	54.4416	0.2049
SVR(rbf t=0.01)	53.4064	0.1907
ANN(logistic(5x10) adam)	38.4773	-1.9223
ANN(identity(8x75) lbfgs)	48.4568	0.2134
ANN(relu(2x10) sgd)	94.7433	-0.2669
ANN(logistic(2x10) sgd)	51.4240	-151.1149
ANN(logistic(5x10) sgd)	52.0099	-182.6213
ANN(relu(8x10) sgd)	53.2530	-161.9513
ANN(logistic(8x10) sgd)	51.4431	-161.5546
ANN(identity(2x25) sgd)	100.3109	-0.8602
ANN(logistic(5x25) sgd)	96.8827	-2.9242
ANN(identity(8x25) sgd)	50.4804	-109.548
ANN(logistic(2x75) sgd)	115.0522	-0.9323
ANN(relu(2x75) sgd)	96.7011	-0.9969

Table 25 - Best results per method and worst five absolute results using the Air Quality Benchmark considering MAPE% and R^2. Periodicity=48h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=2)	74.6896	0.0918
RFR(N=8 n=2)	57.3716	0.0894
KNN(N=5 euclidian)	59.9550	0.0618
KNN(N=8 uniform)	60.8792	0.0876
SVR(linear t=0.01)	51.0538	0.1356
SVR(rbf t=0.001)	53.5310	0.1432
ANN(logistic(5x10) sgd)	54.4294	-195.3038
ANN(logistic(8x10) adam)	42.3600	-16.9236
ANN(logistic(8x10) sgd)	51.7032	-298.6566
ANN(identity(2x25) adam)	55.1116	0.1288
ANN(logistic(2x25) sgd)	45.9128	-212.2232
ANN(identity(5x25) adam)	82.8408	-0.2374
ANN(logistic(8x25) sgd)	46.6322	-187.5112
ANN(identity(5x75) adam)	131.8398	-0.3324
ANN(relu(5x75) lbfgs)	84.0198	-0.0360
ANN(identity(8x75) adam)	92.4474	-0.1442
ANN(identity(8x75) lbfgs)	51.2560	0.1288
ANN(relu(8x75) lbfgs)	83.5472	0.1086
ANN(relu(8x75) sgd)	46.3732	-514.0264

Table 26 - Best results per method and worst five absolute results using the Air Quality Benchmark considering MAPE% and R^2. Periodicity=168h and 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	43.4950	0.3440
KNN(N=5 uniform)	55.7290	0.0840
KNN(N=8 euclidian)	53.1500	-0.0140
SVR(poly t=0.001)	40.8320	0.1300
SVR(rbf t=0.001)	42.0980	0.4450
ANN(identity(2x10) adam)	40.1770	0.6070
ANN(relu(2x10) adam)	30.4780	0.5630
ANN(relu(5x10) sgd)	103.2190	-6.5420
ANN(logistic(2x25) sgd)	43.9140	-38.9150
ANN(identity(5x25) sgd)	102.7010	-0.6680
ANN(relu(8x25) lbfgs)	94.8320	0.0680
ANN(logistic(2x75) sgd)	38.4510	-8.7220
ANN(identity(5x75) adam)	132.8040	-0.0920
ANN(relu(5x75) sgd)	49.9650	-46.0300
ANN(logistic(5x75) sgd)	38.5020	-18.3340
ANN(identity(8x75) adam)	95.0020	-0.1040
ANN(relu(8x75) sgd)	45.4410	-26.4190
ANN(logistic(8x75) sgd)	38.9530	-8.5070

Table 27 - Best results per method and worst five absolute results using the Air Quality Benchmark considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 14 temperature samples.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	53.4922	0.1200
RFR(N=8 n=8)	52.5494	0.0954
KNN(N=8 uniform)	42.0407	0.1905
KNN(N=8 euclidian)	42.5650	0.1944
SVR(linear t=0.01)	37.7116	0.0338
SVR(poly t=0.01)	49.1795	0.1671
ANN(relu(2x10) sgd)	72.3625	0.0037
ANN(logistic(2x10) sgd)	45.0115	-336.8660
ANN(logistic(5x10) sgd)	45.1383	-184.2940
ANN(relu(8x10) sgd)	43.1508	-196.8260
ANN(logistic(8x10) sgd)	45.2175	-459.2080
ANN(relu(2x25) sgd)	38.5479	-2.0600
ANN(relu(5x25) lbfgs)	68.5936	0.1522
ANN(relu(8x25) lbfgs)	91.9585	0.0222
ANN(relu(8x25) sgd)	43.3140	-161.678
ANN(relu(2x75) lbfgs)	60.8049	0.1593
ANN(relu(5x75) lbfgs)	71.1109	-0.1879
ANN(relu(8x75) lbfgs)	71.5410	-0.0122

Table 28 - Best results per method and worst five absolute results using the Air Quality Benchmark considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 1 temperature sample.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	55.4912	0.1028
RFR(N=8 n=2)	50.8083	0.0037
KNN(N=2 euclidian)	57.4636	0.0547
KNN(N=8 euclidian)	52.2677	-0.0798
SVR(linear t=0.01)	37.8870	-0.0066
SVR(poly t=0.01)	44.9880	0.1445
ANN(relu(2x10) adam)	36.0354	-0.3135
ANN(logistic(2x10) sgd)	45.0466	-253.5970
ANN(relu(5x10) lbfgs)	40.4353	0.1668
ANN(logistic(5x10) sgd)	45.0457	-290.2020
ANN(relu(8x10) sgd)	46.0659	-107.7290
ANN(logistic(8x10) sgd)	45.1562	-314.4880
ANN(relu(2x25) sgd)	82.3003	-0.1524
ANN(logistic(2x25) sgd)	43.7625	-105.6350
ANN(relu(5x25) lbfgs)	61.1379	0.0250
ANN(relu(2x75) lbfgs)	60.3620	0.1342
ANN(relu(5x75) lbfgs)	66.9720	0.1515
ANN(relu(8x75) lbfgs)	76.3993	0.0665

Appendix B. Real Cases Extensive Results

Table 29 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 3 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	12.7483	0.7483
KNN(N=8 uniform)	12.2673	0.7451
KNN(N=8 euclidian)	12.3673	0.7459
SVR(linear t=0.01)	14.2473	0.6986
ANN(logistic(2x10) sgd)	46.2568	-145.1720
ANN(logistic(5x10) sgd)	46.5271	-185.6020
ANN(relu(8x10) sgd)	44.5658	-54.4669
ANN(logistic(8x10) sgd)	46.3663	-130.9340
ANN(identity(2x25) sgd)	74.0854	-8.0659
ANN(relu(2x25) sgd)	50.6113	-39.4338
ANN(identity(5x25) sgd)	13.9207	0.7752
ANN(identity(2x75) lbfgs)	11.1476	0.7673
ANN(relu(5x75) sgd)	44.6248	-65.6867

Table 30 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 70 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	13.4106	0.8134
KNN(N=5 euclidian)	11.8821	0.8406
KNN(N=8 euclidian)	11.7590	0.8153
SVR(poly t=0.01)	11.7056	0.8061
SVR(rbf t=0.001)	11.7401	0.8409
ANN(logistic(2x10) sgd)	45.9034	-117.6720
ANN(identity(5x10) sgd)	50.8090	0.4524
ANN(logistic(5x10) sgd)	46.3261	-156.0330
ANN(relu(8x10) sgd)	44.6836	-159.0220
ANN(logistic(8x10) sgd)	46.1331	-140.1020
ANN(identity(2x25) sgd)	47.2779	-0.0333
ANN(relu(5x25) sgd)	46.7739	-152.5060
ANN(identity(8x25) lbfgs)	9.7884	0.8609
ANN(relu(8x25) sgd)	46.8387	-100.8360
ANN(relu(5x75) lbfgs)	10.6036	0.8651

Table 31 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=12h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	11.9181	0.7060
RFR(N=8 n=8)	11.1305	0.6854
KNN(N=8 uniform)	10.8876	0.7150
KNN(N=8 euclidian)	10.9437	0.7168
SVR(linear t=0.01)	11.4832	0.6552
SVR(linear t=0.001)	11.6596	0.6575
ANN(logistic(2x10) sgd)	48.5924	-629.1740
ANN(logistic(5x10) sgd)	48.3851	-832.8060
ANN(relu(8x10) sgd)	48.4048	-174.6750
ANN(logistic(8x10) sgd)	48.7741	-1038.5800
ANN(logistic(2x25) sgd)	47.9611	-181.9480
ANN(relu(8x25) sgd)	47.1044	-463.6900
ANN(identity(2x75) lbfgs)	10.7419	0.7366
ANN(relu(2x75) lbfgs)	10.4128	0.7184
ANN(relu(8x75) sgd)	43.8625	-372.8340

Table 32 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=48h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	13.2595	0.7762
KNN(N=8 euclidian)	13.1200	0.6712
SVR(poly t=0.001)	13.2795	0.7700
ANN(relu(2x10) sgd)	53.1620	0.4772
ANN(logistic(2x10) sgd)	44.5578	-166.0950
ANN(logistic(5x10) sgd)	45.9922	-259.4880
ANN(identity(8x10) lbfgs)	10.7018	0.8322
ANN(relu(8x10) lbfgs)	12.1365	0.8330
ANN(logistic(8x10) sgd)	45.9240	-274.5410
ANN(logistic(8x25) sgd)	45.8570	-138.0270
ANN(relu(2x75) sgd)	46.8510	-15.7305
ANN(identity(5x75) adam)	67.4950	0.3073
ANN(identity(8x75) adam)	55.5892	0.4225
ANN(relu(8x75) sgd)	45.3408	-176.4960

Table 33 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=168h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	13.3250	0.7860
RFR(N=8 n=2)	14.0190	0.7890
KNN(N=8 uniform)	11.0920	0.8190
KNN(N=8 euclidian)	11.0630	0.8180
SVR(linear t=0.001)	12.6800	0.8450
SVR(rbf t=0.01)	11.7100	0.8190
ANN(logistic(2x10) sgd)	45.0430	-17.3460
ANN(relu(5x10) sgd)	40.8100	-34.6400
ANN(logistic(5x10) sgd)	45.6030	-20.6460
ANN(relu(8x10) sgd)	45.2410	-13.4780
ANN(logistic(8x10) sgd)	45.6800	-22.1340
ANN(relu(2x25) adam)	12.0670	0.8700
ANN(identity(5x75) adam)	70.0100	0.1420
ANN(identity(8x75) adam)	52.4430	0.2730
ANN(relu(8x75) sgd)	45.1960	-17.8100

Table 34 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 14 demand samples, with clustering.

Model	MAPE (%)	R^2
RFR(N=8 n=2)	13.4729	0.7837
RFR(N=8 n=8)	13.3262	0.7631
KNN(N=8 uniform)	12.0748	0.7696
SVR(poly t=0.001)	13.8029	0.7502
SVR(rbf t=0.01)	14.6436	0.7519
ANN(logistic(2x10) sgd)	44.9430	-121.8070
ANN(identity(5x10) lbfgs)	11.2067	0.8040
ANN(logistic(5x10) sgd)	46.1839	-157.0520
ANN(logistic(8x10) sgd)	46.2711	-180.4210
ANN(identity(2x25) adam)	18.3966	0.8229
ANN(relu(8x25) sgd)	44.6163	-67.4683
ANN(identity(2x75) adam)	55.2542	-2.2277
ANN(identity(5x75) adam)	58.4519	-1.5973
ANN(identity(8x75) adam)	48.6786	0.2219
ANN(relu(8x75) sgd)	43.9586	-78.9544

Table 35 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 14 temperature samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	11.3892	0.8180
KNN(N=8 uniform)	11.5921	0.7772
KNN(N=8 euclidian)	11.6246	0.7773
SVR(poly t=0.01)	12.2143	0.8264
SVR(poly t=0.001)	12.1394	0.8260
ANN(logistic(2x10) sgd)	46.7262	-144.3600
ANN(relu(5x10) sgd)	48.4879	-83.5033
ANN(logistic(5x10) sgd)	46.3969	-137.8030
ANN(identity(8x10) lbfgs)	10.2219	0.8589
ANN(relu(8x10) sgd)	43.0892	-44.8661
ANN(logistic(8x10) sgd)	46.4429	-159.4570
ANN(relu(2x25) lbfgs)	10.2686	0.8590
ANN(logistic(2x25) sgd)	45.3330	-44.0360

Table 36 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 1 temperature sample.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	11.5031	0.8148
KNN(N=5 euclidian)	11.8994	0.8222
KNN(N=8 euclidian)	11.2073	0.8168
SVR(poly t=0.001)	11.0606	0.8306
ANN(logistic(2x10) sgd)	46.0413	-139.9110
ANN(relu(5x10) sgd)	48.5140	-41.7009
ANN(logistic(5x10) sgd)	46.3528	-135.7060
ANN(relu(8x10) lbfgs)	10.0870	0.8677
ANN(logistic(8x10) sgd)	46.4220	-151.3870
ANN(logistic(5x25) sgd)	45.0743	-44.9706
ANN(logistic(8x25) sgd)	45.1479	-47.8530
ANN(relu(5x75) lbfgs)	9.42810	0.8601

Table 37 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 14 rain occurrence samples.

Model	MAPE (%)	R^2
RFR(N=8 n=2)	11.6163	0.8247
KNN(N=8 uniform)	11.5921	0.7772
KNN(N=8 euclidian)	11.6246	0.7773
SVR(poly t=0.01)	12.2143	0.8264
SVR(poly t=0.001)	12.1394	0.8260
ANN(logistic(2x10) sgd)	46.7262	-144.3600
ANN(relu(5x10) sgd)	48.4879	-83.5033
ANN(logistic(5x10) sgd)	46.3969	-137.8030
ANN(identity(8x10) lbfgs)	10.2219	0.8589
ANN(relu(8x10) sgd)	43.0892	-44.8661
ANN(logistic(8x10) sgd)	46.4429	-159.4570
ANN(relu(2x25) lbfgs)	10.2686	0.8590
ANN(logistic(2x25) sgd)	45.3330	-44.0360

Table 38 - Best results per method and worst five absolute results using the WD2 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 1 rain occurrence sample.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	11.7693	0.8063
RFR(N=8 n=2)	11.8227	0.8190
KNN(N=5 euclidian)	11.8994	0.8222
KNN(N=8 euclidian)	11.2073	0.8168
SVR(poly t=0.001)	11.0606	0.8306
ANN(logistic(2x10) sgd)	46.0413	-139.9110
ANN(relu(5x10) sgd)	48.5140	-41.7009
ANN(logistic(5x10) sgd)	46.3528	-135.7060
ANN(relu(8x10) lbfgs)	10.0870	0.8677
ANN(logistic(8x10) sgd)	46.4220	-151.3870
ANN(logistic(5x25) sgd)	45.0743	-44.9706
ANN(logistic(8x25) sgd)	45.1479	-47.8530
ANN(relu(5x75) lbfgs)	9.4281	0.8601

Table 39 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 3 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	12.0342	0.7697
RFR(N=8 n=2)	11.9156	0.7666
KNN(N=5 euclidian)	12.3124	0.7661
KNN(N=8 euclidian)	12.0094	0.7648
SVR(linear t=0.001)	11.6971	0.7677
ANN(logistic(2x10) sgd)	43.5024	-169.9370
ANN(logistic(5x10) sgd)	44.0668	-221.1870
ANN(relu(8x10) sgd)	41.9927	-66.7910
ANN(logistic(8x10) sgd)	43.8324	-157.8360
ANN(identity(2x25) sgd)	65.6498	-9.6067
ANN(relu(2x25) sgd)	44.6328	-39.0780
ANN(identity(5x25) sgd)	12.7028	0.8226
ANN(identity(8x25) lbfgs)	12.0528	0.7641
ANN(relu(5x75) sgd)	41.0907	-60.5046

Table 40 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 70 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	13.4106	0.8134
KNN(N=5 euclidian)	11.8821	0.8406
KNN(N=8 euclidian)	11.7590	0.8153
SVR(poly t=0.01)	11.7056	0.8061
SVR(rbf t=0.001)	11.7401	0.8409
ANN(logistic(2x10) sgd)	45.9034	-117.6720
ANN(identity(5x10) sgd)	50.8090	0.4524
ANN(logistic(5x10) sgd)	46.3261	-156.0330
ANN(relu(8x10) sgd)	44.6836	-159.0220
ANN(logistic(8x10) sgd)	46.1331	-140.1020
ANN(identity(2x25) sgd)	47.2779	-0.0333
ANN(relu(5x25) sgd)	46.7739	-152.5060
ANN(identity(8x25) lbfgs)	9.7884	0.8609
ANN(relu(8x25) sgd)	46.8387	-100.8360
ANN(relu(5x75) lbfgs)	10.6036	0.8651

Table 41 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=12h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	9.9076	0.7597
RFR(N=8 n=2)	9.7931	0.7006
RFR(N=8 n=8)	9.6316	0.7077
KNN(N=5 euclidian)	9.2686	0.8256
KNN(N=8 euclidian)	8.7154	0.8197
SVR(linear t=0.01)	10.6549	0.7903
SVR(linear t=0.001)	10.6344	0.7899
SVR(poly t=0.01)	10.0771	0.7724
ANN(relu(2x10) adam)	9.0971	0.8401
ANN(logistic(2x10) sgd)	45.2108	-599.1000
ANN(relu(5x10) sgd)	46.3603	-285.5580
ANN(logistic(5x10) sgd)	45.2551	-853.4010
ANN(relu(8x10) lbfgs)	8.3981	0.8084
ANN(logistic(8x10) sgd)	45.3979	-1025.2400
ANN(logistic(5x25) sgd)	44.7174	-229.5330
ANN(relu(8x25) sgd)	43.9322	-336.4700
ANN(relu(8x75) sgd)	40.6193	-471.7230

Table 42 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=48h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	11.6410	0.8308
KNN(N=5 uniform)	11.4342	0.8312
KNN(N=5 euclidian)	11.4730	0.8312
KNN(N=8 euclidian)	11.4170	0.8192
SVR(rbf t=0.001)	10.2100	0.8520
ANN(relu(2x10) sgd)	46.7222	0.5078
ANN(logistic(2x10) sgd)	43.3685	-206.0330
ANN(logistic(5x10) sgd)	44.7705	-302.8730
ANN(logistic(8x10) sgd)	44.8175	-334.4800
ANN(relu(5x25) lbfgs)	9.5678	0.8995
ANN(logistic(8x25) sgd)	43.8787	-162.538
ANN(identity(5x75) adam)	63.6835	0.3532
ANN(identity(8x75) adam)	48.6477	0.3645
ANN(relu(8x75) sgd)	43.7310	-174.5080

Table 43 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=168h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=8 n=8)	12.6710	0.7960
KNN(N=5 euclidian)	12.0610	0.8040
SVR(linear t=0.01)	11.0230	0.8940
SVR(linear t=0.001)	11.0380	0.8940
ANN(relu(2x10) adam)	13.7640	0.9100
ANN(logistic(2x10) sgd)	46.2120	-23.8530
ANN(relu(5x10) sgd)	43.5220	-47.5580
ANN(logistic(5x10) sgd)	47.0510	-29.6210
ANN(logistic(8x10) sgd)	47.1160	-32.5080
ANN(identity(2x25) lbfgs)	10.2320	0.8980
ANN(logistic(5x25) sgd)	46.5880	-16.790
ANN(relu(8x25) sgd)	46.0820	-20.060
ANN(identity(5x75) adam)	60.7830	0.3430
ANN(identity(8x75) adam)	47.3480	0.2090

Table 44 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 14 demand samples, with clustering.

Model	MAPE (%)	R^2
RFR(N=8 n=2)	11.7326	0.8297
RFR(N=8 n=8)	11.0807	0.8244
KNN(N=8 euclidian)	11.6443	0.7750
SVR(rbf t=0.001)	11.3157	0.8181
ANN(logistic(2x10) sgd)	42.6618	-139.1720
ANN(relu(5x10) sgd)	40.0786	-79.9029
ANN(logistic(5x10) sgd)	43.8621	-181.8010
ANN(identity(8x10) sgd)	10.1124	0.8937
ANN(logistic(8x10) sgd)	43.9642	-210.2970
ANN(identity(5x75) adam)	45.8483	-3.0169
ANN(identity(8x75) adam)	51.6922	0.4192
ANN(relu(8x75) sgd)	40.9030	-79.2701

Table 45 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 14 temperature samples.

Model	MAPE (%)	R^2
RFR(N=8 n=2)	9.8030	0.8666
RFR(N=8 n=8)	9.6849	0.8659
KNN(N=2 euclidian)	11.1467	0.8392
KNN(N=5 euclidian)	10.9577	0.8323
SVR(linear t=0.01)	10.7233	0.8889
SVR(linear t=0.001)	10.5579	0.8888
SVR(poly t=0.01)	9.2700	0.8799
ANN(logistic(2x10) sgd)	44.1800	-178.2840
ANN(relu(5x10) sgd)	46.0999	-109.4890
ANN(logistic(5x10) sgd)	43.8648	-169.2130
ANN(identity(8x10) lbfgs)	7.8589	0.9271
ANN(relu(8x10) sgd)	40.8359	-56.7314
ANN(logistic(8x10) sgd)	43.9698	-194.4150
ANN(logistic(2x25) sgd)	42.8944	-54.4253
ANN(relu(2x75) lbfgs)	8.5299	0.9302

Table 46 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 1 temperature sample.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	9.7087	0.8724
RFR(N=8 n=8)	9.3218	0.8689
KNN(N=5 euclidian)	10.1289	0.8646
KNN(N=8 euclidian)	10.2240	0.8661
SVR(poly t=0.001)	8.9696	0.8872
SVR(rbf t=0.001)	9.1003	0.8877
ANN(logistic(2x10) sgd)	43.5989	-168.4150
ANN(relu(5x10) sgd)	46.2823	-58.3954
ANN(logistic(5x10) sgd)	43.8577	-166.6780
ANN(logistic(8x10) sgd)	43.9734	-184.1180
ANN(relu(2x25) lbfgs)	7.9536	0.9210
ANN(logistic(8x25) sgd)	42.7828	-59.9247
ANN(relu(8x75) lbfgs)	8.3372	0.9253

Table 47 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 14 rain occurrence samples.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	53.4922	0.1200
RFR(N=8 n=8)	52.5494	0.0954
KNN(N=8 uniform)	42.0407	0.1905
KNN(N=8 euclidian)	42.5650	0.1944
SVR(linear t=0.01)	37.7116	0.0338
SVR(poly t=0.01)	49.1795	0.1671
ANN(relu(2x10) sgd)	72.3625	0.0037
ANN(logistic(2x10) sgd)	45.0115	-336.8660
ANN(logistic(5x10) sgd)	45.1383	-184.2940
ANN(relu(8x10) sgd)	43.1508	-196.8260
ANN(logistic(8x10) sgd)	45.2175	-459.2080
ANN(relu(2x25) sgd)	38.5479	-2.0600
ANN(relu(5x25) lbfgs)	68.5936	0.1522
ANN(relu(8x25) lbfgs)	91.9585	0.0222
ANN(relu(8x25) sgd)	43.3140	-161.6780
ANN(relu(2x75) lbfgs)	60.8049	0.1593
ANN(relu(5x75) lbfgs)	71.1109	-0.1879
ANN(relu(8x75) lbfgs)	71.5410	-0.0122

Table 48 - Best results per method and worst five absolute results using the WD4 considering MAPE% and R^2. Periodicity=24h, 14 demand samples and 1 rain occurrence sample.

Model	MAPE (%)	R^2
RFR(N=5 n=2)	9.7658	0.8726
RFR(N=8 n=8)	9.4858	0.8677
KNN(N=5 euclidian)	10.1289	0.8646
KNN(N=8 euclidian)	10.2240	0.8661
SVR(poly t=0.001)	8.9696	0.8872
SVR(rbf t=0.001)	9.1003	0.8877
ANN(logistic(2x10) sgd)	43.5989	-168.4150
ANN(relu(5x10) sgd)	46.2823	-58.39540
ANN(logistic(5x10) sgd)	43.8577	-166.6780
ANN(logistic(8x10) sgd)	43.9734	-184.1180
ANN(relu(2x25) lbfgs)	7.9536	0.9210
ANN(logistic(8x25) sgd)	42.7828	-59.9247
ANN(relu(8x75) lbfgs)	8.3372	0.9253

Table 49 - Best results per method and worst five absolute results using the Water Utility 2 Data considering MAPE% and R^2. Periodicity=24h, 3 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	9.9675	0.6773
RFR(N=8 n=8)	9.9732	0.6915
KNN(N=8 uniform)	9.3134	0.7067
SVR(linear t=0.001)	25.2395	-0.5368
ANN(logistic(2x10) sgd)	29.7087	-178.7570
ANN(logistic(5x10) adam)	31.6718	-2.7423
ANN(logistic(5x10) sgd)	29.6264	-206.8540
ANN(logistic(8x10) adam)	31.5534	-3.5034
ANN(logistic(8x10) sgd)	30.0727	-151.9580
ANN(identity(2x25) sgd)	54.5704	-6.7840
ANN(relu(2x25) sgd)	40.2826	-16.5371
ANN(logistic(2x25) sgd)	31.1841	-51.2836
ANN(relu(8x25) sgd)	29.6059	-107.0180
ANN(relu(2x75) lbfgs)	9.9887	0.6818
ANN(relu(5x75) sgd)	29.8539	-135.9530
ANN(identity(8x75) lbfgs)	9.7927	0.6672

Table 50 - Best results per method and worst five absolute results using the Water Utility 2 Data considering MAPE% and R^2. Periodicity=24h, 70 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=8)	9.3039	0.6605
RFR(N=8 n=2)	8.9601	0.6442
KNN(N=2 euclidian)	8.9935	0.5739
KNN(N=8 uniform)	9.2044	0.5986
KNN(N=8 euclidian)	9.0825	0.5958
SVR(linear t=0.01)	25.5959	-0.4830
ANN(logistic(2x10) sgd)	32.7025	-165.2830
ANN(identity(5x10) sgd)	62.0026	-0.5120
ANN(logistic(5x10) sgd)	32.4560	-189.341
ANN(identity(8x10) sgd)	64.2566	-3.4529
ANN(relu(8x10) sgd)	30.0568	-185.8090
ANN(logistic(8x10) sgd)	32.6608	-173.847
ANN(identity(2x25) sgd)	34.7822	-1.1469
ANN(relu(2x25) sgd)	39.8639	-7.4205
ANN(identity(5x25) sgd)	39.0150	-3.7618
ANN(relu(5x25) sgd)	33.3484	-194.2230
ANN(identity(5x75) lbfgs)	8.6002	0.6626
ANN(relu(8x75) lbfgs)	9.4401	0.6874

Table 51 - Best results per method and worst five absolute results using the Water Utility 2 Data considering MAPE% and R^2. Periodicity=12h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=2 n=8)	8.9036	0.6751
RFR(N=8 n=8)	7.9930	0.6619
KNN(N=5 uniform)	8.3411	0.5887
SVR(linear t=0.001)	22.3370	-0.9828
ANN(logistic(2x10) sgd)	30.3468	-471.4890
ANN(identity(5x10) sgd)	34.5003	-4.8133
ANN(relu(5x10) sgd)	29.2088	-705.8190
ANN(logistic(5x10) sgd)	30.4489	-453.0830
ANN(logistic(8x10) sgd)	30.7203	-558.5900
ANN(identity(2x25) lbfgs)	7.8026	0.6314
ANN(relu(2x25) lbfgs)	8.1854	0.6999
ANN(logistic(2x25) lbfgs)	31.6348	-3.9896
ANN(logistic(8x25) adam)	31.5735	-3.1073
ANN(logistic(5x75) adam)	34.2959	-2.5425
ANN(logistic(8x75) adam)	33.5945	-2.7869
ANN(relu(8x75) sgd)	29.1408	-307.7080

Table 52 - Best results per method and worst five absolute results using the Water Utility 2 Data considering MAPE% and R^2. Periodicity=48h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=2)	9.0508	0.6842
RFR(N=8 n=2)	9.0728	0.6852
KNN(N=8 euclidian)	8.3250	0.7040
SVR(linear t=0.001)	25.0966	-0.5912
ANN(relu(2x10) lbfgs)	8.2274	0.7314
ANN(logistic(2x10) sgd)	27.8056	-311.7380
ANN(logistic(5x10) sgd)	28.3334	-334.1980
ANN(logistic(8x10) sgd)	28.2276	-374.4200
ANN(identity(5x25) adam)	40.2604	0.0552
ANN(identity(8x25) adam)	36.4218	-0.1298
ANN(relu(8x25) sgd)	29.4154	-262.5890
ANN(identity(2x75) adam)	40.0262	-0.2684
ANN(identity(5x75) adam)	66.5688	-0.4332
ANN(identity(8x75) adam)	54.9606	-0.0958
ANN(relu(8x75) sgd)	27.6782	-245.9070

Table 53 - Best results per method and worst five absolute results using the Water Utility 2 Data considering MAPE% and R^2. Periodicity=168h, 14 demand samples.

Model	MAPE (%)	R^2
RFR(N=5 n=2)	11.7680	0.5640
KNN(N=2 euclidian)	10.3930	0.6140
SVR(linear t=0.001)	30.5940	-0.7980
SVR(poly t=0.001)	31.0890	-0.7640
ANN(relu(2x10) lbfgs)	10.1170	0.6920
ANN(relu(5x10) sgd)	22.4490	-17.0430
ANN(identity(8x10) sgd)	76.5810	-1.9980
ANN(relu(8x10) sgd)	31.3060	-11.2130
ANN(logistic(8x10) sgd)	27.8370	-13.8550
ANN(identity(5x25) adam)	45.4600	-0.2020
ANN(relu(8x25) sgd)	30.8000	-26.2620
ANN(logistic(8x25) sgd)	31.9570	-24.7830
ANN(identity(5x75) adam)	72.932	-0.0360
ANN(identity(5x75) sgd)	50.7480	-3.3030
ANN(identity(8x75) adam)	59.5450	0.1090

Table 54 - Best results per method and worst five absolute results using the Water Utility 2 Data considering MAPE% and R^2. Periodicity=24h, 14 demand samples, with clustering.

Model	MAPE (%)	R^2
RFR(N=8 n=2)	11.4313	0.6056
RFR(N=8 n=8)	11.2744	0.5958
KNN(N=5 uniform)	10.5019	0.6097
KNN(N=8 uniform)	10.2742	0.6041
SVR(linear t=0.01)	25.0169	-0.4118
ANN(logistic(2x10) sgd)	30.0887	-99.1525
ANN(relu(5x10) sgd)	26.3775	-77.5403
ANN(logistic(5x10) sgd)	30.6563	-95.9168
ANN(identity(8x10) lbfgs)	7.9235	0.7420
ANN(relu(8x10) sgd)	31.2555	-25.5605
ANN(logistic(8x10) sgd)	30.9117	-117.1810
ANN(logistic(5x25) sgd)	31.3586	-29.3283
ANN(relu(8x25) sgd)	31.0995	-54.7944
ANN(logistic(8x25) sgd)	31.4429	-30.6580
ANN(identity(2x75) adam)	33.8954	-0.7640
ANN(relu(8x75) sgd)	29.5566	-77.0348